

Robert Meersman
Zahir Tari
Pilar Herrero (Eds.)

LNCS 5333

On the Move to Meaningful Internet Systems: OTM 2008 Workshops

OTM Confederated International Workshops and Posters
ADI, AWeSoMe, COMBEK, EI2N, IWSSA, MONET,
OnToContent + QSI, ORM, PerSys, RDDS, SEMELS, and SWWS 2008
Monterrey, Mexico, November 2008, Proceedings

 Springer

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

University of Dortmund, Germany

Madhu Sudan

Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max-Planck Institute of Computer Science, Saarbruecken, Germany

Robert Meersman Zahir Tari
Pilar Herrero (Eds.)

On the Move to Meaningful Internet Systems: OTM 2008 Workshops

OTM Confederated International Workshops and Posters
ADI, AWeSoMe, COMBEK, EI2N, IWSSA, MONET,
OnToContent+QSI, ORM, PerSys, RDDS, SEMELS,
and SWWS 2008
Monterrey, Mexico, November 9-14, 2008
Proceedings



Springer

المنار
للإستشارات

Volume Editors

Robert Meersman
Vrije Universiteit Brussel (VUB), STARLab
Bldg G/10, Pleinlaan 2, 1050, Brussels, Belgium
E-mail: meersman@vub.ac.be

Zahir Tari
RMIT University, School of Computer Science and Information Technology
Bld 10.10, 376-392 Swanston Street, VIC 3001, Melbourne, Australia
E-mail: zahir.tari@rmit.edu.au

Pilar Herrero
Universidad Politécnica de Madrid, Facultad de Informática
Campus de Montegancedo S/N, 28660 Boadilla del Monte, Madrid, Spain
E-mail: pherrero@fi.upm.es

Library of Congress Control Number: 2008938152

CR Subject Classification (1998): H.2, H.3, H.4, C.2, H.5, I.2, D.2, K.4

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

ISSN 0302-9743
ISBN-10 3-540-88874-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-88874-1 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media
springer.com

© Springer-Verlag Berlin Heidelberg 2008
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12556069 06/3180 5 4 3 2 1 0

Volume Editors

Robert Meersman
Zahir Tari
Pilar Herrero

ADI

Stefan Jablonski
Olivier Curé
Christoph Bussler

AWeSoMe

Jörg Denzinger
Pilar Herrero
Gonzalo Méndez
Rainer Unland

COMBEK

Pieter De Leenheer
Martin Hepp
Amit Sheth

EI2N

Arturo Molina
Andrew Kusiak
Hervé Panetto
Peter Bernus

IWSSA

Lawrence Chung
José Luis Garrido
Nary Subramanian
Manuel Noguera

MONET

Fernando Ferri
Irina Kondratova
Arianna D'Ulizia
Patrizia Grifoni

OnToContent & QSI

Andreas Schmidt
Mustafa Jarrar
Ramon F. Brena
Francisco Cantu-Ortiz

ORM

Terry Halpin
Sjir Nijssen

PerSys

Skevos Evripidou
Roy Campbell

RDDS

Achour Mostefaoui
Eiko Yoneki

SEMELS

Elena Simperl
Reto Krummenacher
Lyndon Nixon
Emanuele Della Valle
Ronaldo Menezes

SWWS

Tharam S. Dillon

OTM 2008 General Co-chairs' Message

Dear OnTheMove Participant, or Reader of these Proceedings,

The OnTheMove 2008 event in Monterrey, Mexico, 9–14 November, further consolidated the growth of the conference series that was started in Irvine, California in 2002, and held in Catania, Sicily in 2003, in Cyprus in 2004 and 2005, in Montpellier in 2006, and in Vilamoura in 2007. The event continues to attract a diversifying and representative selection of today's worldwide research on the scientific concepts underlying new computing paradigms, which, of necessity, must be distributed, heterogeneous and autonomous yet meaningfully collaborative.

Indeed, as such large, complex and networked intelligent information systems become the focus and norm for computing, there continues to be an acute and increasing need to address and discuss in an integrated forum the implied software, system and enterprise issues as well as methodological, semantical, theoretical and applicational issues. As we all know, email, the Internet, and even video conferences are not sufficient for effective and efficient scientific exchange. The OnTheMove (OTM) Federated Conferences series has been created to cover the scientific exchange needs of the community/ies that work in the broad yet closely connected fundamental technological spectrum of data and web semantics, distributed objects, web services, databases, information systems, enterprise workflow and collaboration, ubiquity, interoperability, mobility, grid and high-performance computing.

OnTheMove aspires to be a primary scientific meeting place where all aspects for the development of such Internet- and Intranet-based systems in organizations and for e-business are discussed in a scientifically motivated way. This sixth edition of the OTM Federated Conferences event again provided an opportunity for researchers and practitioners to understand and publish these developments within their individual as well as within their broader contexts.

Originally the federative structure of OTM was formed by the co-location of three related, complementary and successful main conference series: DOA (Distributed Objects and Applications, since 1999), covering the relevant infrastructure-enabling technologies, ODBASE (Ontologies, DataBases and Applications of SEmantics, since 2002) covering Web semantics, XML databases and ontologies, and CoopIS (Cooperative Information Systems, since 1993) covering the application of these technologies in an enterprise context through e.g., workflow systems and knowledge management. In 2006 a fourth conference, GADA (Grid computing, high-performAnce and Distributed Applications) was added to this as a main symposium, and last year the same happened with IS (Information Security). Both of these started as successful workshops at OTM, the first covering the large-scale integration of heterogeneous computing systems and data resources with the aim of providing a global computing space,

the second covering the issues of security in complex Internet-based information systems.

Each of these five conferences encourages researchers to treat their respective topics within a framework that incorporates jointly (a) theory, (b) conceptual design and development, and (c) applications, in particular case studies and industrial solutions.

Following and expanding the model created in 2003, we again solicited and selected quality workshop proposals to complement the more “archival” nature of the main conferences with research results in a number of selected and more “avant-garde” areas related to the general topic of distributed computing. For instance, the so-called Semantic Web has given rise to several novel research areas combining linguistics, information systems technology, and artificial intelligence, such as the modeling of (legal) regulatory systems and the ubiquitous nature of their usage. We were glad to see that in spite of OnTheMove switching sides of the Atlantic, seven of our earlier successful workshops (notably AweSOMe, SWWS, ORM, OnToContent, MONET, PerSys, RDDS) re-appeared in 2008 with a third or even fourth edition, sometimes by alliance with other newly emerging workshops, and that no fewer than seven brand-new independent workshops could be selected from proposals and hosted: ADI, COMBEK, DiSCo, IWSSA, QSI and SEMELS. Workshop audiences productively mingled with each other and with those of the main conferences, and there was considerable overlap in authors. The OTM organizers are especially grateful for the leadership, diplomacy and competence of Dr. Pilar Herrero in managing this complex and delicate process for the fifth consecutive year.

Unfortunately however in 2008 the number of quality submissions for the OnTheMove Academy (formerly called Doctoral Consortium Workshop), our “vision for the future” in research in the areas covered by OTM, proved too low to justify a 2008 edition in the eyes of the organizing faculty. We must however thank Antonia Albani, Sonja Zaplata and Johannes Maria Zaha, three young and active researchers, for their efforts in implementing our interactive formula to bring PhD students together: research proposals are submitted for evaluation; selected submissions and their approaches are (eventually) to be presented by the students in front of a wider audience at the conference, and intended to be independently and extensively analyzed and discussed in public by a panel of senior professors. Prof. Em. Jan Dietz, the Dean of the OnTheMove Academy, also is stepping down this year, but OnTheMove is committed to continuing this formula with a new Dean and peripatetic faculty.

All five main conferences and the associated workshops shared the distributed aspects of modern computing systems, and the resulting application-pull created by the Internet and the so-called Semantic Web. For DOA 2008, the primary emphasis stayed on the distributed object infrastructure; for ODBASE 2008, it has become the knowledge bases and methods required for enabling the use of formal semantics; for CoopIS 2008, the focus as usual was on the interaction of such technologies and methods with management issues, such as occur in networked organizations, for GADA 2008, the main topic was again the scalable integration

of heterogeneous computing systems and data resources with the aim of providing a global computing space, and in IS 2008 the emphasis was on information security in the networked society. These subject areas overlapped in a scientifically natural fashion and many submissions in fact also treated an envisaged mutual natural impact among them. As for the earlier editions, the organizers wanted to stimulate this cross-pollination by a *shared* program of famous keynote speakers: this year we were proud to announce Dan Atkins of the U.S. National Science Foundation and the University of Michigan, Hector Garcia-Molina of Stanford, Rick Hull of IBM T.J. Watson Lab, Ted Goranson of Sirius-Beta and of Paradigm Shift, and last but not least Cristina Martinez-Gonzalez of the European Commission with a special interest in more future scientific collaboration between the EU and Latin America, as well as emphasizing the concrete outreach potential of new Internet technologies for enterprises anywhere.

This year the registration fee structure strongly encouraged multiple event attendance by providing *all* main conference authors with free access or discounts to *all* other conferences or workshops (workshop authors paid a small extra fee to attend the main conferences). In both cases the price for these combo tickets was made lower than in 2007 in spite of the higher organization costs and risks!

We received a total of 292 submissions for the five main conferences and 171 submissions in total for the 14 workshops. The numbers are about 30% lower than for 2007, which was not unexpected because of the transatlantic move of course, and the emergent need to establish the OnTheMove brand in the Americas, a process that will continue as we proceed in the coming years. But, not only may we indeed again claim success in attracting an increasingly representative volume of scientific papers, many from US, Central and South America already, but these numbers of course allow the program committees to compose a high-quality cross-section of current research in the areas covered by OTM. In fact, in spite of the larger number of submissions, the Program Chairs of each of the three main conferences decided to accept only approximately the same number of papers for presentation and publication as in 2006 and 2007 (i.e., average 1 paper out of 3-4 submitted, not counting posters). For the workshops, the acceptance rate varies but the aim was to stay as strict as before, consistently about 1 accepted paper for 2-3 submitted. We have separated the proceedings into three books with their own titles, two for the main conferences and one for the workshops, and we are grateful to Springer for their suggestions and collaboration in producing these books and CDROMs. The reviewing process by the respective program committees was again performed very professionally, and each paper in the main conferences was reviewed by at least three referees, with arbitrated email discussions in the case of strongly diverging evaluations. It may be worthwhile emphasizing that it is an explicit OnTheMove policy that all conference program committees and chairs make their selections completely autonomously from the OTM organization itself. The OnTheMove Federated Event organizers again made all proceedings available on a CDROM to all

participants of conferences resp. workshops, independently of their registration to a specific conference resp. workshop. Paper proceedings were on request this year, and incurred an extra charge.

The General Chairs were once more especially grateful to the many people directly or indirectly involved in the setup of these federated conferences. Few people realize what a large number of people have to be involved, and what a huge amount of work, and in 2008 certainly also financial risk, the organization of an event like OTM entails. Apart from the persons in their roles mentioned above, we therefore in particular wish to thank our 17 main conference PC co-chairs:

GADA 2008	Dennis Gannon, Pilar Herrero, Daniel Katz, María S. Pérez
DOA 2008	Mark Little, Alberto Montresor, Greg Pavlik
ODBASE 2008	Malu Castellanos, Fausto Giunchiglia, Feng Ling
CoopIS 2008	Johann Eder, Masaru Kitsuregawa, Ling Liu
IS 2008	Jong Hyuk Park, Bart Preneel, Ravi Sandhu, André Zúquete
50 Workshop PC Co-chairs	Stefan Jablonski, Olivier Curé, Christoph Bussler, Jörg Denzinger, Pilar Herrero, Gonzalo Méndez, Rainer Unland, Pieter De Leenheer, Martin Hepp, Amit Sheth, Stefan Decker, Ling Liu, James Caverlee, Ying Ding, Yihong Ding, Arturo Molina, Andrew Kusiak, Hervé Panetto, Peter Bernus, Lawrence Chung, José Luis Garrido, Nary Subramanian, Manuel Noguera, Fernando Ferri, Irina Kondratova, Arianna D'ulizia, Patrizia Grifoni, Andreas Schmidt, Mustafa Jarrar, Terry Halpin, Sjir Nijssen, Skevos Evripidou, Roy Campbell, Anja Schanzenberger, Ramon F. Brena, Hector Ceballos, Yolanda Castillo, Achour Mostefaoui, Eiko Yoneki, Elena Simperl, Reto Krummenacher, Lyndon Nixon, Emanuele Della Valle, Ronaldo Menezes, Tharam S. Dillon, Ernesto Damiani, Elizabeth Chang, Paolo Ceravolo, Amandeep S. Sidhu

All, together with their many PC members, did a superb and professional job in selecting the best papers from the large harvest of submissions.

We must all be grateful to Ana Cecilia Martinez-Barbosa for researching and securing the local and sponsoring arrangements on-site, to Josefa Kumpfmüller for many useful scientific insights in the dynamics of our transatlantic move, and to our extremely competent and experienced Conference Secretariat and technical support staff in Antwerp, Daniel Meersman, Ana-Cecilia (again), and Jan Demey, and last but not least to our apparently never-sleeping Melbourne Program Committee Support Team, Vidura Gamini Abhaya and Anshuman Mukherjee.

The General Chairs gratefully acknowledge the academic freedom, logistic support and facilities they enjoy from their respective institutions, Vrije Universiteit Brussel (VUB) and RMIT University, Melbourne, without which such an enterprise would not be feasible.

We do hope that the results of this federated scientific enterprise contribute to your research and your place in the scientific network... We look forward to seeing you again at next year's event!

August 2008

Robert Meersman
Zahir Tari

Organization Committee

OTM (On The Move) is a federated event involving a series of major international conferences and workshops. These proceedings contain the papers presented at the OTM 2008 Federated Workshops, consisting of the following workshops: ADI 2008 (1st International Workshop on Ambient Data Integration), AWeSoMe 2008 (4th International Workshop on Agents and Web Services Merging in Distributed Environments), COMBEK 2008 (1st International Workshop on Community-Based Evolution of Knowledge-Intensive Systems), EI2N 2008 (3rd International Workshop on Enterprise Integration, Interoperability and Networking), IWSSA 2008 (7th International Workshop on System/Software Architectures), MONET 2008 (3rd International Workshop on MOBILE and NETworking Technologies for social applications), OnToContent 2008 (3rd International Workshop on Ontology Content and Evaluation in Enterprise) + QSI 2008 (1st International Workshop on Quantitative Semantic methods for the Internet), ORM 2008 (International Workshop on Object-Role Modeling), PerSys 2008 (3rd International Workshop on Pervasive Systems), RDDS 2008 (3rd International Workshop on Reliability in Decentralized Distributed Systems), SEMELS 2008 (1st International Workshop on Semantic Extensions to Middleware: Enabling Large Scale Knowledge Applications), and SWWS 2008 (4th International IFIP Workshop on Semantic Web and Web Semantics).

Executive Committee

OTM 2008 General Co-chairs	Robert Meersman (Vrije Universiteit Brussel, Belgium) and Zahir Tari (RMIT University, Australia)
OTM 2008 Workshop Co-chairs	Pilar Herrero (Universidad Politécnica de Madrid, Spain) and Robert Meersman (Vrije Universiteit Brussel, Belgium)
ADI 2008 PC Co-chairs	Stefan Jablonski (University of Bayreuth, Germany), Olivier Curé (Université Paris Est, France) and Christoph Bussler (Merced Systems Inc., USA)
AWeSoMe 2008 PC Co-chairs	Jörg Denzinger (Department of Computer Science, Canada), Pilar Herrero (Universidad Politécnica de Madrid, Spain), Gonzalo Méndez (Facultad de Informática, Spain), and Rainer Unland (Institute for Computer Science and Business Information Systems, Germany)

COMBEK 2008 PC Co-chairs	Pieter De Leenheer (Vrije Universiteit Brussel, Belgium), Martin Hepp (Bundeswehr University, Germany), and Amit Sheth (Wright State University, USA)
EI2N 2008 PC Co-chairs	Arturo Molina (ITESM, Mexico), Andrew Kusiak (University of Iowa, USA), Hervé Panetto (University of Nancy, France) and Peter Bernus (Griffith University, Australia)
IWSSA 2008 PC Co-chairs	Lawrence Chung (University of Texas at Dallas, USA), José Luis Garrido (University of Granada, Spain), Nary Subramanian (University of Texas at Tyler, USA), and Manuel Noguera (University of Granada, Spain)
MONET 2008 PC Co-chairs	Fernando Ferri (National Research Council, Italy), Irina Kondratova (National Research Council, Italy), Arianna D’ulizia (National Research Council, Italy), and Patrizia Grifoni (National Research Council, Italy)
OnToContent & QSI 2008 PC Co-chairs	Andreas Schmidt (FZI Research Center for Information Technologies, Germany), Mustafa Jarrar (University of Cyprus, Cyprus), Ramon F. Brena (Tec de Monterrey, Mexico), and Francisco Cantu-Ortiz (Tec de Monterrey, Mexico)
ORM 2008 PC Co-chairs	Terry Halpin (Neumont University, USA) and Sjir Nijssen (PNA, The Netherlands)
PerSys 2008 PC Co-chairs	Skevos Evripidou (University of Cyprus, Cyprus) and Roy Campbell (University of Illinois at Urbana-Champaign, USA)
RDDS 2008 PC Co-chairs	Achour Mostefaoui (IRISA/Université Rennes 1, France) and Eiko Yoneki (University of Cambridge, UK)
SEMELS 2008 PC Co-chairs	Elena Simperl (University of Innsbruck, Austria), Reto Kruppenacher (University of Innsbruck, Austria), Lyndon Nixon (Free University Berlin, Germany), Emanuele Della Valle (Politecnico di Milano, Italy), and Ronaldo Menezes (Florida Institute of Technology, USA)
SWWS 2008 PC Co-chairs	Tharam S. Dillon (Curtin University of Technology, Australia), Ernesto Damiani (Milan University, Italy), and Elizabeth Chang (Curtin University of Technology, Australia)
Publication Co-chairs	Vidura Gamini Abhaya (RMIT University, Australia) and Anshuman Mukherjee (RMIT University, Australia)

Local Organizing Chair	Lorena G. Gómez Martínez (Tecnológico de Monterrey, Mexico)
Conferences Publicity Chair	Keke Chen (Yahoo!, USA)
Workshops Publicity Chair	Gonzalo Mendez (Universidad Complutense de Madrid, Spain)
Secretariat	Ana-Cecilia Martinez Barbosa, Jan Demey, and Daniel Meersman

ADI (Ambient Data Integration) 2008 Program Committee

Christoph Bussler	Myriam Lamolle
Olivier Curé	Richard Lenz
Mathieu D'Aquin	Sascha Mueller
Wolfgang Deiters	Erich Ortner
Stefan Jablonski	Gerhard Rambold
Robert Jeansoulin	Riccardo Rosati
Roland Kaschek	Kurt Sandkuhl

AWeSoMe (International Workshop on Agents and Web Services Merging in Distributed Environments) 2008 Program Committee

Mohsen Afsharchi	Michael Maximilien
M. Brian Blake	Barry Norton
José Luis Bosque	Julian Padget
Juan A. BotíaBlaya	Mauricio Paletta
Ramon Brena	José Peña
Scott Buffett	María Pérez
Paul Buhler	Manuel Salvadores
Blanca Caminero	Alberto Sánchez
Jose Cardoso	Candelaria Sansores
Adam Cheyer	Marius-Calin Silaghi
Ian Dickinson	Santtu Toivonen
Maria Fasli	Julita Vassileva
Roberto A. Flores	Yao Wang
Dominic Greenwood	Chengqi Zhang
Jingshan Huang	Henry Tirri
Dan Marinescu	Ángel Lucas González Martínez
Gregorio Martínez	Antonio Garcia Dopico
Viviana Mascardi	

COMBEK (International Workshop on Community-Based Evolution of Knowledge-Intensive Systems) 2008 Program Committee

Hugo Liu	Igor Mozetic
Natalya Noy	Davide Eynard
Munindar Singh	Tanguy Coenen
Dragan Gasevic	Stijn Christiaens
Juan-Carlos Fernandez-Ramil	Katharina Siorpaes
Christopher Thomas	Marta Sabou
Andreas Schmidt	Denny Vrandecic
Alicia Diaz	Konstantinos Kotis
Tom Mens	Valentin Zacharias
Mark Aakhus	Siegfried Handschuh
Filippo Lanubile	John Breslin
Aldo de Moor	

EI2N (International Workshop on Enterprise Integration, Interoperability and Networking) Program Committee

Giuseppe Berio	Jörg Müller
Peter Bernus	Ovidiu Noran
Nacer Boudjlida	Angel Ortiz
David Chen	Hervé Panetto
Vincent Chapurlat	Jin Woo Park
Michele Dassisti	Li Qin
Ricardo Goncalves	Aurelian Mihai Stanescu
Roland Jochem	Janusz Szpytko
Andrew Kusiak	Bruno Vallespir
Juan-Carlos Mendez	François B. Vernadat
Shimon Nof	George Weichhart
Vidosav D. Majstorovich	Lawrence Whitman
Ivan Mezgar	Martin Zelm
Arturo Molina	Xuan Zhou

IWSSA (International Workshop on System/Software Architectures) Program Committee

Philippe Aniorde	Francois Coallier
Hernán Astudillo	Kendra Cooper
Doo-Hwan Bae	Rafael Corchuelo
Jaelson Castro	Lirong Dai
Roger Champagne	Sergiu Dascalu

Yannis A. Dimitriadis
 Jing Dong
 Jesús Favela
 Juan Fernández-Ramil
 Paul Gruenbacher
 Lars Grunske
 Fred Harris
 Michael Hinchey
 Mara V. Hurtado
 Stan Jarzabek
 Li Jiang
 Carlos Juiz
 Rick Kazman
 Pericles Loucopoulos
 María D. Lozano
 Chung-Horng Lung
 Stephen J. Mellor

Tommi Mikkonen
 Masaki Murakami
 Sergio F. Ochoa
 Patricia Paderewski
 Sooyong Park
 Óscar Pastor
 Fabio Paternò
 Juan Pavón
 María Luisa Rodríguez
 Gustavo Rossi
 Vespe Savikko
 Michael Shin
 Yeong Tae Song
 Sebastian Uchitel
 Roel Wieringa
 Andrea Zisman

OnToContent + QSI 2008 Program Committee

Ernst Biesalski
 Thanasis Bouras
 Simone Braun
 Christopher Brewster
 Michael Brown
 Yannis Charalabidis
 Ernesto Damiani
 Gouvas Panagiotis
 Guizzardi Giancarlo
 Mohand-Said Hacid
 Martin Hepp
 Stijn Heymans
 Christine Kunzmann
 Stefanie Lindstaedt
 Tobias Ley
 Clementina Marinoni
 Alessandro Oltramari
 Viktoria Pammer
 Paul Piwek
 Christophe Roche
 Peter Scheir
 Miguel-Angel Sicilia
 Barry Smith

Armando Stellato
 Sergio Tessaris
 Robert Tolksdorf
 Francky Trichet
 Vervenne Luk
 Miguel Alonso Pardo
 Jerome Euzenat
 Sara Garza
 Randy Goebel
 Adolfo Guzman
 Graeme Hirst
 Fakhri Karray
 Richard Kittredge
 Ana Maguitman
 Trevor Martin
 Antonio Moreno
 Vivi Nastase
 Eduardo Ramirez
 Vasile Rus
 Elie Sanchez
 Juan M. Torres Moreno
 Manuel Vilares
 Hugo Zaragoza

ORM (International Workshop on Object-Role Modeling) 2008 Program Committee

Roel Baardman
Guido Bakema
Herman Balsters
Linda Bird
Anthony Bloesch
Scott Becker
Peter Bollen
Lex Bruil
Andy Carver
Don Baisley
Donald Chapin
Dave Cuyler
Olga De Troyer
Jan Dietz
Gordon Everest

Ken Evans
John Hall
Pat Hallock
Hank Hermans
Stijn Hoppenbrouwers
Mike Jackson
Mustafa Jarrar
Elisa Kendall
Mark Linehan
Inge Lemmens
Bodil Madsen
Robert Meersman
Tony Morgan
Maurice Nijssen
Anita Nuopponen

PerSys (International Workshop on Pervasive Systems) 2008 Program Committee

Susana Alcalde Baguees
Xavier Alamn Roldan
Jalal Al-Muhtadi
Christian Becker
Michael Beigl
Alastair Beresford
Antonio Coronato
Thanos Demiris
Hakan Duman
Alois Ferscha
Nikolaos Georgantas
Patricia Grifoni
Alex Healing
Bob Hulsebosch
Sergio Ilarri

Cornel Klein
Nik Klever
Irina Kondratova
Andrew Rice
Philip Robinson
George Samaras
Gregor Schiele
Behrooz Shirazi
Sotirios Terzis
Verena Tuttlies
Valerie Issarny
Gregory Yovanof
Apostolos Zarras
Arkady Zaslavsky

RDDS (International Workshop on Reliability in Decentralized Distributed Systems) Program Committee

Licia Capra
Paolo Costa

Simon Courtenage
Patrick Eugster

Ludger Fiege
 Seth Gilbert
 Christos Gkantsidis
 Eli Katsiri
 Michael Kounavis
 Marco Mamei
 Jonathan Munson

Maziar Nekovee
 Andrea Passarella
 Peter Pietzuch
 Matthieu Roy
 Francois Taiani
 Ruediger Kapitza

SEMELS (International Workshop on Semantic Extensions to Middleware: Enabling Large Scale Knowledge Applications) Program Committee

Andrea Omicini
 Carlos Pedrinaci
 Daniel Wutke
 David Robertson
 Doug Foxvog
 Franco Zambonelli
 Ian Oliver
 Ilya Zaihrayeu
 Jacek Kopecky
 Lee Feigenbaum

Manfred Bortenschlager
 Robert Tolksdorf
 Chen Wei
 Mourad Ouzzani
 Babak Esfandiari
 Wojciech Barczynski
 Angelo Brayner
 Alexandre C.T. Vidal
 Carole Goble
 Sonia Bergamaschi

SWWS (International IFIP Workshop on Semantic Web and Web Semantics) Program Committee

Aldo Gangemi
 Amit Sheth
 Angela Schwering
 Avigdor Gal
 Carlos Sierra
 Carole Goble
 Chris Bussler
 Claudia d'Amato
 David Bell
 Elena Camossi
 Elisa Bertino
 Elizabeth Chang
 Ernesto Damiani
 Farookh Hussain
 Feng Ling

Grigoris Antoniou
 Hai Zhuge
 Han Jaiwei
 John Debenham
 John Mylopoulos
 Katia Sycara
 Krzysztof Janowicz
 Kokou Yetongnon
 Kyu-Young Whang
 Ling Liu
 Lizhu Zhou
 Lotfi Zadeh
 Manfred Hauswirth
 Maria Andrea Rodríguez-Tastets
 Masood Nikvesh

OTM Workshops 2008 Additional Reviewers

Abdul-Rahman Mawlood-Yunis	Irina Kondratova
Adolfo Guzman	Janusz Szpytko
Alex Healing	Jinwoo Park
Alexandre Vidal	John Hall
Andrew Kusiak	Jos Vos
Andy Phippen	Joy Garfield
Angel Ortiz	Juan Manuel Torres
Anita Nuopponen	Julita Vassileva
Antonio Garcia Dopico	Ken Evans
Antonio Moreno	Lawrence Whitman
Arturo Molina	Lex Bruil
Baba Piprani	Linda Bird
Behrooz Shirazi	Lirong Dai
Bodil Nistrup Madsen	Luigi Gallo
Bruno Vallespir	Manish Bhide
C.-C. Jay Kuo	Manuel Resinas
Carlos Guerrero	Marco Padula
Christopher Thomas	Mark Aakhus
Dan Marinescu	Martin Zelm
Dave Robertson	Maurice Nijssen
David Cuyler	Mauricio Paletta
Domenico Gendarmi	Mehdi Khouja
Dominic Greenwood	Michael Brown
Don Baisley	Miguel-Angel Sicilia
Donald Chapin	Mihai Lintean
Douglas Foxvog	Mikael Wiberg
Eduardo Ramirez Rangel	Mike Jackson
Elie Sanchez	Mohsen Afsharchi
Ernst Biesalski	Myriam Lamolle
Fernanda Alencar	Olivier Curé
François Coallier	Patrick Hallock
François Vernadat	Patrick van Bommel
Frederick Harris	Peter Bernus
Gerhard Skagestein	Peter Spyns
Gobinda Chowdhury	R.A.U. Juchter van Bergen Quast
Gordon Everest	Randy Goebel
Graeme Hirst	Ricardo Goncalves
Gregory Yovanof	Riccardo Martoglia
Hank Hermans	Robert Jeansoulin
Herman Balsters	Roberto Flores
Ian Oliver	Roel Baardman
In-Gwon Song	Roland Jochem
Irina Kondratova	Roy Campbell

Sang-Uk Jeon
 Sara Elena Garza Villarreal
 Scot Becker
 Sebastian Kruk
 Sergio Ochoa
 Sergiu Dascalu
 Sheila Kinsella
 Simon Courtenage
 Sinjae Kang
 Sotirios Terzis
 Stephen Mellor

Thanassis Bouras
 Thanos Demiris
 Tim Strayer
 Tommo Reti
 Tu Peng
 Vespe Savikko
 Vidosav Majstorovic
 Xuan Zhou
 Yajing Zhao
 Yuri Tijerino

Sponsoring Institutions

OTM 2008 was proudly sponsored by BN (Bauch & Navratil, Czech Republic), Nuevo Leon, and the City of Monterrey.



Supporting Institutions

OTM 2008 was proudly supported by RMIT University (School of Computer Science and Information Technology), Vrije Universiteit Brussel (Department of Computer Science), Technical University of Monterrey and Universidad Politécnica de Madrid.



Vrije Universiteit Brussel



TECNOLÓGICO
DE MONTERREY.



Table of Contents

Posters of the 2008 CoopIS (Cooperative Information Systems) International Conference

Real-Time Reasoning Based on Event-Condition-Action Rules	1
<i>Ying Qiao, Xiang Li, Hongan Wang, and Kang Zhong</i>	
PASS It ON (PASSION): An Adaptive Online Load-Balancing Algorithm for Distributed Range-Query Specialized Systems	3
<i>Ioannis Konstantinou, Dimitrios Tsoumakos, and Nectarios Koziris</i>	
Dynamic Source Selection to Handle Changes of User's Interest in Continuous Query	6
<i>Kosuke Ohki, Yousuke Watanabe, and Hiroyuki Kitagawa</i>	
The (Similarity) Matrix Reloaded	8
<i>Avigdor Gal</i>	
Enabling Flexible Execution of Business Processes	10
<i>S. Jablonski, M. Faerber, F. Jochaud, M. Götz, and M. Iglér</i>	
Collaborative Environment for Engineering Simulations with Integrated VR Visualization	12
<i>Ismael H.F. Santos, Alberto B. Raposo, and Marcelo Gattass</i>	
A Method for Searching Keyword-Lacking Files Based on Interfile Relationships	14
<i>Tetsutaro Watanabe, Takashi Kobayashi, and Haruo Yokota</i>	
Really Simple Security for P2P Dissemination of Really Simple Syndication	16
<i>Anwitaman Datta and Liu Xin</i>	
Mining and Analyzing Organizational Social Networks Using Minimum Spanning Tree	18
<i>Victor Ströele A. Menezes, Ricardo Tadeu da Silva, Moisés Ferreira de Souza, Jonice Oliveira, Carlos E.R. de Mello, Jano Moreira de Souza, and Geraldo Zimbrão</i>	
MADIK: A Collaborative Multi-agent ToolKit to Computer Forensics	20
<i>Bruno W.P. Hoelz, Célia G. Ralha, Rajiv Geeverghese, and Hugo C. Junior</i>	
Modularizing Monitoring Rules in Business Processes Models	22
<i>Oscar González, Rubby Casallas, and Dirk Deridder</i>	

An Optimal Approach for Workflow Staff Assignment Based on Hidden Markov Models	24
<i>Hedong Yang, Chaokun Wang, Yingbo Liu, and Jianmin Wang</i>	
Behavioral Compatibility of Web Services	27
<i>Zhangbing Zhou, Sami Bhiri, Walid Gaaloul, Lei Shu, and Manfred Hauswirth</i>	
A Reverse Order-Based QoS Constraint Correction Approach for Optimizing Execution Path for Service Composition	29
<i>Kaijun Ren, Nong Xiao, Jinjun Chen, and Junqiang Song</i>	

Posters of the 2008 ODBASE (Ontologies, DataBases, and Applications of Semantics) International Conference

Data Mining of Specific-Domain Ontology Components	31
<i>J.R.G. Pulido, M.A. Aréchiga, and M.E.C. Espinosa</i>	
Distributed Data Mining by Means of SQL Enhancement	34
<i>Marcin Gorawski and Ewa Pluciennik</i>	
Construction and Querying of Relational Schema for Ontology Instances Data	36
<i>Maciej Falkowski and Czeslaw Jedrzejek</i>	
Evaluation of the Navigation through Image Parts in the ImageNotion Application	38
<i>Andreas Walter, Gabor Nagypal, and Simone Braun</i>	
Instance-Based Ontology Matching Using Regular Expressions	40
<i>Katrin Zaiß, Tim Schlüter, and Stefan Conrad</i>	

Workshop on Ambient Data Integration (ADI)

ADI 2008 PC Co-chairs' Message	43
--	----

Architectures

Modelling ETL Processes of Data Warehouses with UML Activity Diagrams	44
<i>Lilia Muñoz, Jose-Norberto Mazón, Jesús Pardillo, and Juan Trujillo</i>	
Implementing Conceptual Data Integration in Process Modeling Methodologies for Scientific Applications	54
<i>Bernhard Volz</i>	
Theoretical and Practical Challenges of Integrating Ecosystem Data	64
<i>M. Hauhs, B. Trancón y Widemann, and O. Archner</i>	

Methods for Data Integration

The Health Problems of Data Integration	65
<i>Avigdor Gal</i>	
Computing Path Similarity Relevant to XML Schema Matching	66
<i>Amar Zerdazi and Myriam Lamolle</i>	
Improving Search and Navigation by Combining Ontologies and Social Tags	76
<i>Silvia Bindelli, Claudio Criscione, Carlo A. Curino, Mauro L. Drago, Davide Eynard, and Giorgio Orsi</i>	

Workshop on Agents and Web Services Merging in Distributed Environments (AWeSoMe)

AWESOME 2008 PC Co-chairs' Message	87
--	----

Agents, Grid and Applications

Teaching about Madrid: A Collaborative Agents-Based Distributed Learning Course	88
<i>José Luis Bosque, Pilar Herrero, and Susana Mata</i>	
The Principle of Immanence in GRID-Multiagent Integrated Systems . . .	98
<i>Pascal Dugenie, Clement Jonquet, and Stefano A. Cerri</i>	
MASD: Towards a Comprehensive Multi-agent System Development Methodology	108
<i>T. Abdelaziz, M. Elammari, and C. Branki</i>	
A Mobile Device Based Multi-agent System for Structural Optimum Design Applications	118
<i>Cherif Branki, Tilmann Bitterberg, and Hanno Hildmann</i>	

Agent Communication and Coordination

Discovering Pragmatic Similarity Relations between Agent Interaction Protocols	128
<i>Maricela Bravo and José Velazquez</i>	
On the Relevance of Organizational Structures for a Technology of Agreement	138
<i>Holger Billhardt, Roberto Centeno, Alberto Fernández, Ramón Hermoso, Rubén Ortiz, Sascha Ossowski, and Matteo Vasirani</i>	

Learning, Information Exchange, and Joint-Deliberation through
Argumentation in Multi-agent Systems 150
Santi Ontañón and Enric Plaza

Web Services and SOA

A User-Centric Service Composition Approach 160
Gabriela Vulcu, Sami Bhiri, Manfred Hauswirth, and Zhangbing Zhou

Towards Reliable SOA – An Architecture for Quality Management of
Web Services 170
Ingo J. Timm and Thorsten Scholz

**Workshop on Community-Based Evolution of
Knowledge-Intensive Systems (COMBEK)**

COMBEK 2008 PC Co-chairs’ Message 181

Early Afternoon Session

Semi-automated Consensus Finding for Meaning Negotiation 183
Christophe Debruyne, Johannes Peeters, and Allal Zakaria Arrassi

On Supporting HCOME-3O Ontology Argumentation Using Semantic
Wiki Technology 193
Konstantinos Kotis

Morning Session

Inferring Social Groups Using Call Logs 200
Santi Phithakkitnukoon and Ram Dantu

Group Recommendation System for Facebook 211
Enkh-Amgalan Baatarjav, Santi Phithakkitnukoon, and Ram Dantu

Late Afternoon Session

Towards a Scalable and Collaborative Information Integration Platform
and Methodology 220
Felix Van de Maele and Alicia Díaz

Methodological Approach to Determine Appropriately Annotated
Resources in Narrow Folksonomies 230
Céline Van Damme, Stijn Christiaens, and Damien Trog

IFAC/IFIP Workshop on Enterprise Integration, Interoperability and Networking (EI2N)

EI2N 2008 PC Co-chairs' Message	239
---------------------------------------	-----

Enterprise Networking

Business-IT Alignment Domains and Principles for Networked Organizations: A Qualitative Multiple Case Study	241
<i>Roberto Santana Tapia, Maya Daneva, Pascal van Eck, Nicté-Há Castro Cárdenas, and Leida van Oene</i>	
The View-Constraint Duality in Database Systems, Software Engineering, and Systems Engineering	253
<i>John A. Springer and Edward L. Robertson</i>	

Enterprise Integration and Interoperability

Mining Reference Process Models and Their Configurations	263
<i>Florian Gottschalk, Wil M.P. van der Aalst, and Monique H. Jansen-Vullers</i>	
Interoperability Maturity Models – Survey and Comparison	273
<i>Wided Guédria, Yannick Naudet, and David Chen</i>	
Using the Zachman Framework to Achieve Enterprise Integration Based-on Business Process Driven Modelling	283
<i>Javier Espadas, David Romero, David Concha, and Arturo Molina</i>	

Enterprise Modelling and Service Oriented Architecture

Enterprise Modelling Based Services Orchestration	294
<i>Qing Li, Canqiang Li, and Yun Wang</i>	
Service Oriented Architecture vs. Enterprise Architecture: Competition or Synergy?	304
<i>Ovidiu Noran and Peter Bernus</i>	
SCEP-SOA: An Applicative Architecture to Enhance Interoperability in Multi-site Planning	313
<i>Karim Ishak, Bernard Archimède, and Philippe Charbonnaud</i>	

Workshop on System/Software Architectures (IWSSA)

IWSSA 2008 PC Co-chairs' Message	323
--	-----

Workshop Introduction – Requirements and Architectural Design

Semantic-Aided Interactive Identification of Reusable NFR Knowledge Fragments	324
<i>Claudia López, Hernán Astudillo, and Luiz Marcio Cysneiros</i>	
Aspect-Oriented Modeling of Quality Attributes	334
<i>Mónica Pinto and Lidia Fuentes</i>	
Improving Security of Oil Pipeline SCADA Systems Using Service-Oriented Architectures	344
<i>Nary Subramanian</i>	

Evaluation

An Architecture to Integrate Automatic Observation Mechanisms for Collaboration Analysis in Groupware	354
<i>Rafael Duque, María Luisa Rodríguez, María Visitación Hurtado, Manuel Noguera, and Crescencio Bravo</i>	
A Case Study on Architectural Maturity Evaluation: Experience in the Consumer Electronics Domain	364
<i>Kangtae Kim</i>	
Evaluation of an Agent Framework for the Development of Ambient Computing	374
<i>Marcela D. Rodríguez and Jesús Favela</i>	

Architectural Design Based on Components, Services and Patterns

Defining Re-usable Composite Aspect Patterns: An FDAF Based Approach	384
<i>Kun Tian, Kendra M.L. Cooper, Kunwu Feng, and Yan Tang</i>	
Implementation Variants of the Singleton Design Pattern	396
<i>Krzysztof Stencel and Patrycja Węgrzynowicz</i>	
Semantic Interactions for Context-Aware and Service-Oriented Architecture	407
<i>Mehdi Khouja, Carlos Juiz, Ramon Puigjaner, and Farouk Kamoun</i>	
ReWiSe: A New Component Model for Lightweight Software Reconfiguration in Wireless Sensor Networks	415
<i>Amirhosein Taherkordi, Frank Eliassen, Romain Rouvoy, and Quan Le-Trung</i>	

Development

Tackling Automotive Challenges with an Integrated RE and Design Artifact Model	426
<i>Birgit Penzenstadler</i>	
A Q-Learning-Based On-Line Planning Approach to Autonomous Architecture Discovery for Self-managed Software	432
<i>Dongsun Kim and Sooyong Park</i>	
Developing Collaborative Modeling Systems Following a Model-Driven Engineering Approach	442
<i>Jesús Gallardo, Crescencio Bravo, and Miguel Á. Redondo</i>	

Arquitectural Design Based on Components, Services and Frameworks

Assessing Component's Behavioral Interoperability Concerning Goals . . .	452
<i>Weimin Ma, Lawrence Chung, and Kendra Cooper</i>	
A Reference Architecture for Automated Negotiations of Service Agreements in Open and Dynamic Environments	463
<i>Manuel Resinas, Pablo Fernández, and Rafael Corchuelo</i>	
Towards Checking Architectural Rules in Component-Based Design	473
<i>Sebastian Herold</i>	

Workshop on MOBILE and NETWORKING Technologies for SOCIAL Applications (MONET)

MONET 2008 PC Co-chairs' Message	479
--	-----

Challenges and New Approaches for Social Networking

Personal Sphere Information, Histories and Social Interaction between People on the Internet	480
<i>Maria Chiara Caschera, Fernando Ferri, and Patrizia Grifoni</i>	
Discovery of Social Groups Using Call Detail Records	489
<i>Huiqi Zhang and Ram Dantu</i>	
PALASS: A Portable Application for a Location-Aware Social System	499
<i>Martin Press, Daniel Goodwin, and Roberto A. Flores</i>	

Studies and Applications of Multimodality

Toward the Development of an Integrative Framework for Multimodal Dialogue Processing	509
<i>Arianna D’Ulizia, Fernando Ferri, and Patrizia Grifoni</i>	
A Comparison of Microphone and Speech Recognition Engine Efficacy for Mobile Data Entry	519
<i>Joanna Lumsden, Scott Durling, and Irina Kondratova</i>	
A GRID Approach to Providing Multimodal Context-Sensitive Social Service to Mobile Users	528
<i>Massimo Magaldi, Roberto Russo, Luca Bevilacqua, Stefania Pierno, Vladimiro Scotto di Carlo, Fabio Corvino, Luigi Romano, Luisa Capuano, and Ivano De Furio</i>	

Theories and Applications of Mobile Technology in Different Social Contexts

A Comparison of Pseudo-paper and Paper Prototyping Methods for Mobile Evaluations	538
<i>Joanna Lumsden and Ryan MacLean</i>	
A Model for Checking the Integrity Constraints of Mobile Databases . . .	548
<i>Hamidah Ibrahim, Zarina Dzolkhifli, and Praveen Madiraju</i>	
Usability Issues of e-Learning Systems: Case-Study for Moodle Learning Management System	561
<i>Miroslav Minović, Velimir Štavljanin, Miloš Milovanović, and Dušan Starčević</i>	
CPL: Enhancing Mobile Phone Functionality by Call Predicted List . . .	571
<i>Santi Phithakkitnukoon and Ram Dantu</i>	

Workshop on Ontology Content and Evaluation in Enterprise(OnToContent)+ Quantitative Semantic Methods for the Internet (QSI)

OnToContent+QSI 2008 PC Co-chairs’ Message	583
--	-----

Benefits of Ontologies and Human Factors

Measuring the Benefits of Ontologies	584
<i>Tobias Bürger and Elena Simperl</i>	
Towards a Human Factors Ontology for Computer-Mediated Systems . . .	595
<i>Panagiotis Germanakos, Mario Belk, Nikos Tsianos, Zacharias Lekkas, Constantinos Mourlas, and George Samaras</i>	

Ontology Development in Collaborative Networks as a Process of Social Construction of Meaning	605
<i>Carla Pereira and António Lucas Soares</i>	

Folksonomies and Community Aspects

Toward a Community Vision Driven Topical Ontology in Human Resource Management	615
<i>Damien Trog, Stijn Christiaens, Gang Zhao, and Johanna de Laaf</i>	
Automatic Profiling System for Ranking Candidates Answers in Human Resources	625
<i>Rémy Kessler, Nicolas Béchet, Mathieu Roche, Marc El-Bèze, and Juan Manuel Torres-Moreno</i>	

Elements for Quantitative Semantics

Semantic Expansion of Service Descriptions	635
<i>Christian Sánchez and Leonid Sheremetov</i>	
Tuning Topical Queries through Context Vocabulary Enrichment: A Corpus-Based Approach	646
<i>Carlos M. Lorenzetti and Ana G. Maguitman</i>	
Measuring Heterogeneity between Web-Based Agent Communication Protocols	656
<i>Maricela Bravo and José Velazquez</i>	

Workshop on Object-Role Modeling (ORM)

ORM 2008 PC Co-chairs' Message	667
--	-----

Service Orientation

A Metamodel for Enabling a Service Oriented Architecture	668
<i>Baba Piprani, Chong Wang, and Keping He</i>	
Service-Oriented Conceptual Modeling	678
<i>Peter Bollen</i>	

Temporal Modeling and Dynamic Rules

Temporal Modeling and ORM	688
<i>Terry Halpin</i>	
Formal Semantics of Dynamic Rules in ORM	699
<i>Herman Balsters and Terry Halpin</i>	

Fact-Orientation and SBVR

Fact Orientation and SBVR: The Catalyst for Efficient and Fully Integrated Education	709
<i>Jos Vos</i>	
SBVR: A Fact-Oriented OMG Standard	718
<i>Peter Bollen</i>	

Requirements Analysis and Quality Assurance

An Adaptable ORM Metamodel to Support Traceability of Business Requirements across System Development Life Cycle Phases	728
<i>Baba Piprani, Marlena Borg, Josée Chabot, and Éric Chartrand</i>	
Requirements Specification Using Fact-Oriented Modeling: A Case Study and Generalization	738
<i>Gabor Melli and Jerre McQuinn</i>	
A Model for Data Quality Assessment	750
<i>Baba Piprani and Denise Ernst</i>	

Verbalization Issues

Verbalization for Business Rules and Two Flavors of Verbalization for Fact Examples	760
<i>Maurice Nijssen and Inge Lemmens</i>	
How to Avoid Redundant Object-References	770
<i>Andy Carver</i>	

Advanced Constraints and Tool Demos

A Closer Look at the Join-Equality Constraint	780
<i>Gerhard Skagstein and Ragnar Normann</i>	

Using Fact-Oriented Modeling Tools

Model Ontological Commitments Using ORM ⁺ in T-Lex	787
<i>Yan Tang and Damien Trog</i>	
DOGMA-MESS: A Tool for Fact-Oriented Collaborative Ontology Evolution	797
<i>Pieter De Leenheer and Christophe Debruyne</i>	
Evaluation and Enhancements for NORMA: Student User Suggestions	807
<i>Gordon C. Everest</i>	

Workshop on Pervasive Systems (PerSys)

PerSys 2008 PC Co-chairs' Message	819
---	-----

PerSys Opening and Workshop Keynote

Learning, Prediction and Mediation of Context Uncertainty in Smart Pervasive Environments	820
<i>Sajal K. Das and Nirmalya Roy</i>	

Middleware and Applications

Implicit Middleware: A Ubiquitous Abstract Machine	830
<i>T. Riedel, M. Beigl, M. Berchtold, C. Decker, and A. Puder</i>	
Uncertainty Management in a Location-Aware Museum Guide	841
<i>Pedro Damián-Reyes, Jesús Favela, and Juan Contreras-Castillo</i>	
Modeling Context Life Cycle for Building Smarter Applications in Ubiquitous Computing Environments	851
<i>Hyunjun Chang, Seokkyoo Shin, and Changshin Chung</i>	

Frameworks and Platforms

Game Development Framework Based Upon Sensors and Actuators	861
<i>Ray van Brandenburg, Arie Horst, Bas Burgers, and Nirvana Meratnia</i>	
HTTPStream Platform – Low Latency Data for the Web	873
<i>Marios Tziakouris and Paraskevas Evripidou</i>	
Password Streaming for RFID Privacy	883
<i>Victor K.Y. Wu and Roy H. Campbell</i>	

Workshop on Reliability in Decentralized Distributed Systems (RDDS)

RDDS 2008 PC Co-chairs' Message	893
---------------------------------------	-----

Peer-to-Peer

Reliable Peer-to-Peer Semantic Knowledge Sharing System	894
<i>Abdul-Rahman Mawlood-Yunis</i>	
How to Improve the Reliability of Chord?	904
<i>Jacek Cichoń, Andrzej Jasiński, Rafał Kapelko, and Marcin Zawada</i>	

Distributed Algorithms

A Performance Evaluation of g -Bound with a Consistency Protocol Supporting Multiple Isolation Levels	914
<i>R. Salinas, F.D. Muñoz-Escoí, J.E. Armendáriz-Iñigo, and J.R. González de Mendivil</i>	
Integrity Dangers in Certification-Based Replication Protocols	924
<i>M.I. Ruiz-Fuertes, F.D. Muñoz-Escoí, H. Decker, J.E. Armendáriz-Iñigo, and J.R. González de Mendivil</i>	

Workshop on Semantic Extensions to Middleware: Enabling Large Scale Knowledge (SEMELS)

SEMELS 2008 PC Co-chairs' Message	935
---	-----

Extending Middleware with Semantics

Towards Semantically Enhanced Peer-to-Peer File-Sharing	937
<i>Alan Davoust and Babak Esfandiari</i>	
Efficient Content Location in Massively Distributed Triplespaces	947
<i>Kia Teymourian and Lyndon Nixon</i>	
Extending ESB for Semantic Web Services Understanding	957
<i>Antonio J. Roa-Valverde and José F. Aldana-Montes</i>	

Applications of Semantically Extended Middleware

Distributed Workflows: The OpenKnowledge Experience	965
<i>Paolo Besana, Vivek Patkar, David Glasspool, and Dave Robertson</i>	
SD-Core: A Semantic Middleware Applied to Molecular Biology	976
<i>Ismael Navas-Delgado, Amine Kerzazi, Othmane Chniber, and José F. Aldana-Montes</i>	
Towards Knowledge in the Cloud	986
<i>Davide Cerrì, Emanuele Della Valle, David De Francisco Marcos, Fausto Giunchiglia, Dalit Naor, Lyndon Nixon, Kia Teymourian, Philipp Obermeier, Dietrich Rebholz-Schuhmann, Reto Krümmenacher, and Elena Simperl</i>	

Workshop On Semantic Web and Web Semantics (SWWS)

SWWS 2008 PC Co-chairs' Message	997
---	-----

SWWS I

A Deductive Approach for Resource Interoperability and Well-Defined Workflows	998
<i>Nadia Yacoubi Ayadi, Zoé Lacroix, and Maria-Esther Vidal</i>	
Ontology Robustness in Evolution.....	1010
<i>Paolo Ceravolo, Ernesto Damiani, and Marcello Leida</i>	
Learning to Get the Value of Quality from Web Data.....	1018
<i>Guzmán Llambías, Regina Motz, Federico Toledo, and Simon de Uvarow</i>	
Building the Semantic Utility with Standards and Semantic Web Services.....	1026
<i>Mathias Uslar, Sebastian Rohjans, Stefan Schulte, and Ralf Steinmetz</i>	

SWWS II

TICSA Approach: Five Important Aspects of Multi-agent Systems	1036
<i>Maja Hadzic and Elizabeth Chang</i>	
Applications of the ACGT Master Ontology on Cancer	1046
<i>Mathias Brochhausen, Gabriele Weiler, Luis Martín, Cristian Cocos, Holger Stenzhorn, Norbert Graf, Martin Dörr, Manolis Tsiknakis, and Barry Smith</i>	

SWWS III

An Ontology-Based Crawler for the Semantic Web	1056
<i>Felix Van de Maele, Peter Spyns, and Robert Meersman</i>	
Consensus Emergence from Naming Games in <i>Representative Agent</i> Semantic Overlay Networks	1066
<i>Gabriele Gianini, Ernesto Damiani, and Paolo Ceravolo</i>	
A Semantic Crawler Based on an Extended CBR Algorithm	1076
<i>Hai Dong, Farookh Khadeer Hussain, and Elizabeth Chang</i>	
Author Index	1087

Real-Time Reasoning Based on Event-Condition-Action Rules

Ying Qiao, Xiang Li, Hongan Wang, and Kang Zhong

Institute of Software, Chinese Academy of Sciences, No. 4, Zhong Guan Cun South Fourth Street, Hai Dian District
100190 Beijing, China
qiaoying@ios.cn, xiangli05@iscas.ac.cn,
{wha, zhank}@iel.iscas.ac.cn

Abstract. In this paper, we develop a reasoning mechanism for real-time and active database (RTADB). The core of this reasoning mechanism is a real-time inference algorithm based on Event-condition-action (ECA) rules, called RTIAE. This algorithm is based on heuristic search on a rule graph so as to give the actions responding to occurring events as many as possible under certain timing constraint. Furthermore, we conduct a series of simulations studies to analyze the performance of the proposed algorithm and compare it with depth-first algorithm.

Keywords: Real-time and active database, Reasoning, Event-condition-action Rules, Heuristic search, Rule graph.

1 Introduction

A reasoning mechanism is required in real-time and active databases (RTADB) to support the intelligence of real-time applications. An ad-hoc method should be developed to realize this reasoning mechanism rather than using existing techniques of reasoning based on production rules since they cannot describe the active behaviors in the system. Meanwhile, the reasoning mechanism in RTADB is based on event-condition-action (ECA) rules and should obtain the reasoning results in a real-time manner. However, dealing with the real-time issue for the reasoning based on ECA rules is a great challenge and has not been investigated in current research [1-3]. Thus, we present a novel approach to realize a reasoning mechanism based on ECA rules in the real-time and active database. The core of our approach is a real-time inference algorithm based on ECA rules, called RTIAE. It will find actions taken to react to occurring events as many as possible under given timing constraint.

2 Real-Time Inference Algorithm – RTIAE

RTIAE is based on a rule graph which is a directed graph whose vertexes (nodes) represent events, conditions or actions in ECA rules. Two nodes are connected via a directed edge if one of them needs to transfer information to another. The vertexes

that have no incoming edges are called entrance nodes. They represent primitive events occurring in the system. The vertices that have no outgoing edges are called exit nodes. They represent actions. In RTIAE, the reasoning is accomplished via heuristic search on the rule graph. The purpose of heuristic search is to find a path from a specific entrance node to an exit node so that the time consumed for traveling along this path is as short as possible. The search starts from a specific entrance node. During the search, the expected path will be expanded with a node selected via the value of the heuristic function. The backtracking may occur when necessary.

3 Simulation Studies

In the simulations, we use *reasoning success ratio* as a metric to evaluate the performance of the inference algorithm. *Reasoning success ratio* is defined as the ratio of number of actions found to respond to the occurring events to the number of events arrived at the system within a given deadline. We analyze reasoning success ratio of RTIAE and compare it with depth-first search algorithm. The simulation results demonstrate that the RTIAE takes advantages over depth-first algorithm in term of reasoning success ratio for variation of several parameters, i.e., the laxity and the number of events occurring in the system.

4 Conclusions

To support the intelligence of real-time applications, a reasoning mechanism based on ECA rules should be developed for real-time and active database. Since the RTADB should react to external events within certain deadline, the corresponding timing constraints will be imposed on the reasoning process. In this paper, we present a real-time inference algorithm based on ECA rules – RTIAE which exploits the heuristic search on the rule graph to accomplish the reasoning. Furthermore, we conduct a series of simulation studies to analyze the performance of RTIAE.

Acknowledgments. This work is supported by France Telecom (Grant No. 46135653).

References

- 1 Chakravarthy, S., Le, R., Dasari, R.: ECA Rule Processing in Distributed and Heterogeneous Environments. In: International Symposium on Distributed Objects and Applications, pp. 330–335. IEEE Press, New York (1999)
- 2 Li, X., Chapa, S.V., Marin, J.M., Cruz, J.M.: An Application of Conditional Colored Petri Nets: Active Database System. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 4885–4890. IEEE Press, New York (2004)
- 3 White, W., Riedewald, M., Gehrke, J., Demers, A.: What is “next” in event processing? In: 26th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 263–272. ACM Press, New York (2007)

PASS It ON (PASSION): An Adaptive Online Load-Balancing Algorithm for Distributed Range-Query Specialized Systems

Ioannis Konstantinou, Dimitrios Tsoumakos, and Nectarios Koziris

Computing Systems Laboratory
School of Electrical and Computer Engineering
National Technical University of Athens
{fikons,dtsouma,nkozirisg}@cslab.ece.ntua.gr

1 Introduction

A basic requirement for every P2P system is fault-tolerance. Since the primary objective is resource location and sharing, we require that this basic operation takes place in a reliable manner. In a variety of situations with skewed data accesses (e.g., [1], etc) the demand for content can become overwhelming for certain serving peers, forcing them to reject connections. In many cases, these skewed distributions take extreme forms: *Flash crowds*, regularly documented surges in the popularity of certain content, are also known to cause severe congestion and degradation of service [2]. Data replication techniques is one commonly utilized solution to remedy these situations. Nevertheless, there are cases in which the requested resources cannot be arbitrarily replicated. Distributed data-structures that support range-queries is such an example: The keys are stored in the network nodes so that a natural order is preserved. These structures can be very useful in a variety of situations: On-line games, web servers, data-warehousing, etc. In such cases, adaptive and on-line load-balancing schemes must be employed in order to avoid resource unavailability and performance in a variety of workloads [3,4].

Our contribution. In this work, we present *PASSION*, an on-line, adaptive load balancing algorithm that operates on distributed range-partitioned data structures. Our algorithm operates in a completely decentralized manner and requires no kind of global coordination. Its goal is, through key exchange between neighboring nodes according to their current load and individual thresholds, to counterbalance the inequality in load that affects performance. Each peer, upon sensing an overload situation relative to its individual threshold, requests help and proactively sheds a suitable part of its load to its neighbors. Load moves in a “wave-like” fashion from more to less loaded regions of the structure adaptively.

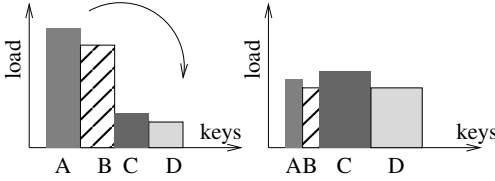
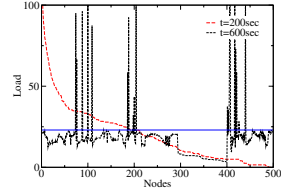
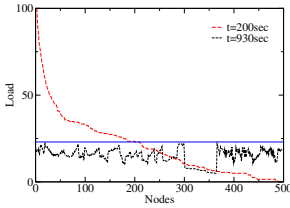
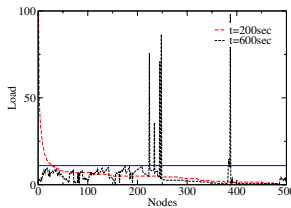
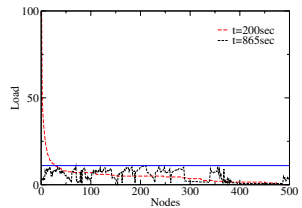


Fig. 1. PASSION example

Fig. 2. Before balance, $\theta = 1$ Fig. 3. After balance, $\theta = 1$ Fig. 4. Before balance, $\theta = 1.4$ Fig. 5. After balance, $\theta = 1.4$

2 Passion

The main idea behind *PASSION* is the following: When the current load of a node exceeds its self-imposed threshold $thresh_i$, the node sends a *HELPREQUEST* message containing its current load to one of its neighbours. The recipient node takes over a portion of the overloaded node's key range. This procedure is performed online, that is, nodes continue to serve requests during the key transfer. The recipient then estimates his new load and if this is above its local threshold, it initiates a new *HELPREQUEST* towards another neighbouring node. The procedure continues *TTL* hops away at most or until all nodes have successfully shed their load below their thresholds.

In order to calculate the portion of load that the overloaded node needs to shed, we introduce the *overThresh* threshold, where $overThresh_i > thresh_i$. If the *splitter's* load is above the *overThresh*, then only a fraction a of the extra load is accepted. Otherwise, the *splitter's* excessive load is fully accepted. Like the simple *thresh*, *overThresh* is a local per-node setting. Its purpose is to smooth out the key/load exchanges between sequential *PASSION* executions.

3 Initial Results

We present an initial simulation-based evaluation of our method. We assume a network size of 500 nodes, all of which are randomly chosen to initiate queries at any given time. More specific, we apply *passion* on our simulator with load generated by: a zipfian distribution for $\theta = 1$ and $\theta = 1.4$ (see Figures 2-5).

References

1. Cha, M., Kwak, H., Rodriguez, P., Ahn, Y., Moon, S.: I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In: IMC 2007. Proceedings of the 7th ACM SIGCOMM conference on Internet measurement (2007)
2. Jung, J., Krishnamurthy, B., Rabinovich, M.: Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites. In: WWW (2002)
3. Karger, D.R., Ruhl, M.: Simple efficient load-balancing algorithms for peer-to-peer systems. *Theory of Computing Systems* 39, 787–804 (2006)
4. Ganesan, P., Bawa, M., Garcia-Molina, H.: Online balancing of range-partitioned data with applications to peer-to-peer systems. In: Proceedings of the Thirtieth international conference on Very large data bases, vol. 30, pp. 444–455 (2004)

Dynamic Source Selection to Handle Changes of User's Interest in Continuous Query

Kosuke Ohki¹, Yousuke Watanabe², and Hiroyuki Kitagawa^{1,3}

¹ Graduate School of Systems and Information Engineering, University of Tsukuba

² Global Scientific Information and Computing Center, Tokyo Institute of Technology

³ Center for Computational Sciences, University of Tsukuba

ohki@kde.cs.tsukuba.ac.jp, watanabe@de.cs.titech.ac.jp,

kitagawa@cs.tsukuba.ac.jp

1 Introduction

The volume of stream data delivered from information sources has been increasing. A demand for efficient processing of stream data has become more and more important. Stream processing systems [1] can continuously process stream data according to user's requests. A request is usually specified as a continuous query written in SQL-like language.

In conventional frameworks, the user must specify information sources in advance, and the user cannot change information sources during query processing. However, there are many cases in which target information sources the user is most interested in change over time. We therefore need an additional framework to deal with changes of the target information sources for the same query.

We propose dynamic source selection for this purpose. Our contributions are as follows: (1) **Proposal of ASSIGN operator to switch information sources dynamically**, (2) **Development of a connection management scheme to reduce connections to unnecessary information sources**, (3) **Implementation of the proposed framework on our stream processing system**, and (4) **Experiment to measure performance in stream environments**.

2 Dynamic Source Selection

We consider an application to track video data of moving objects as a sample case in which target information sources change over time. We assume many network cameras, a database that contains information on locations of the cameras, and location sensors attached to moving objects.

The query processing in our proposed framework is illustrated in Fig. 1. When the system receives the current location of the target object "A", it finds IDs of cameras near the target "A". In this example, the system is receiving a tuple with the camera ID "Camera1". The connection management unit makes a connection to "Camera1" to obtain its video data. The ASSIGN operator then selects "Camera1" and gets the video data from "Video" in information sources "Camera1". The ASSIGN operator outputs a tuple attached with the "Video"

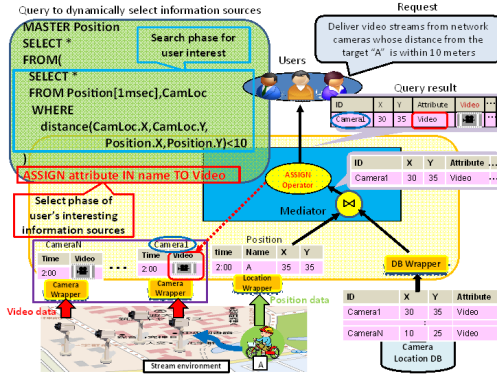


Fig. 1. Tracking application in the proposed framework to dynamically select target information sources

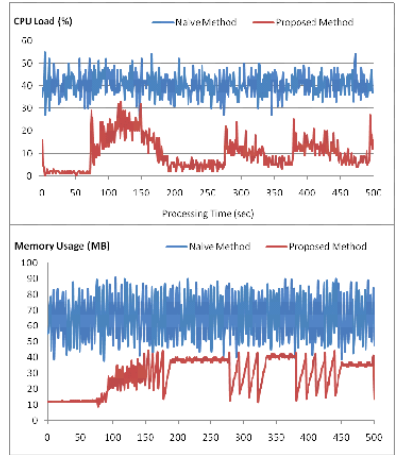


Fig. 2. CPU Load and Memory usage in comparison of naive and proposed methods

attribute. The system repeats this process every time the current location of the target object arrives.

We have implemented the proposed framework in our stream processing system called as StreamSpinner [2].

3 Experiment

We prepared ten cameras and location data of a thousand people in the virtual environment. We compare system loads of the proposed method and a naive method. The naive method here means that the system makes connections to all running network cameras. The result is Fig. 2. We can see our proposed method contributes to reduce system resource consumptions.

Acknowledgments

This research has been supported in part by Japan Science and Technology Agency (CREST), the Grant-in-Aid for Scientific Research from JSPS (#1820005) and MEXT (#19024006).

References

1. Abadi, D., et al.: Aurora: a new model and architecture for data stream management. VLDB Journal 12(2), 120–139 (2003)
2. <http://www.streamspinner.org/>

The (Similarity) Matrix Reloaded

Avigdor Gal

Technion - Israel Institute of Technology
Haifa
32000 Israel
avigal@ie.technion.ac.il

Schema matching provides correspondences between concepts describing the meaning of data in various heterogeneous, distributed data sources. Schema matching is a basic operation of data and schema integration and thus has a great impact on its outcome. The outcome of the matching process can serve in tasks of targeted content delivery, view integration, database integration, query rewriting over heterogeneous sources, duplicate data elimination, and automatic streamlining of workflow activities that involve heterogeneous data sources. As such, schema matching has impact on numerous modern applications from various application areas. It impacts business, where company data sources continuously realign due to changing markets. It also impacts the way business and other information consumers seek information over the Web. Finally, it impacts life sciences, where scientific workflows cross system boundaries more often than not.

Schema matching research has been going on for more than 25 years now (see surveys [1,10,9,11] and online lists, *e.g.*, OntologyMatching¹ and Ziegler²), first as part of schema integration and then as a standalone research. Over the years, a significant body of work was devoted to the identification of *schema matchers*, heuristics for schema matching. Examples include COMA [2], OntoBuilder [5], Similarity Flooding [7], Clio [8], Glue [3], and others. The main objective of schema matchers is to provide schema matchings that will be effective from the user point of view yet not disastrously expensive. Such research has evolved in different research communities, yielding overlapping, similar, and sometimes identical results. Recent benchmarks (such as OAEI³) indicate that the performance of schema matchers still leaves something to be desired.

Somewhat surprisingly, after more than 25 years of research, the research area of schema matching is still borrowing isolated bits and pieces of research from other areas. Some of these efforts have been successful while others have proven to be useful in limited domains only. A theoretical basis allows better design of schema matchers, enhancing user effectiveness.

We promote the use of matrix theory as a theoretical generic foundation to schema matching and propose to adopt the similarity matrix abstraction as a basic data model for schema matching to abstract away the differences between schema matchers and focus on their similarities instead. This approach is useful in designing generic tools with wide applicability. Therefore, basic matching operations will

¹ <http://www.ontologymatching.org/>

² <http://www.ifi.unizh.ch/~piegler/IntegrationProjects.html>

³ <http://www.om2006.ontologymatching.org/OAEI06/directory.htm>

be captured as matrix operations, regardless of whether the matcher itself uses a linguistic heuristic, a machine learning heuristic, *etc.* We argue that such a framework would allow an efficient assessment of schema matcher quality, yielding a better mechanism for designing and applying new schema matchers.

Using this data model we propose to conceptually separate schema matchers into first line and second line matchers. First line matchers are designed as application of existing works in other areas (*e.g.*, machine learning) to schemata. Second line matchers operate on the outcome of other schema matchers to improve their original outcome. Existing examples of second line matchers include similarity flooding [6], and schema matching verification [4]. We claim that this classification, together with the use of similarity matrix as a data model, benefits the design of new schema matchers.

Given the practical need for automating schema matching activities, and the limitations of the state-of-the-art foundations, the significance of developing a common theoretical generic foundation for schema matching design is apparent. This new perspective of existing representations advances the state-of-the-art in providing new mechanisms to deal with the ever daunting problem of schema matching.

References

1. Batini, C., Lenzerini, M., Navathe, S.: A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys* 18(4), 323–364 (1986)
2. Do, H., Rahm, E.: COMA - a system for flexible combination of schema matching approaches. In: *Proceedings of the International conference on Very Large Data Bases (VLDB)*, pp. 610–621 (2002)
3. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to map between ontologies on the semantic web. In: *Proceedings of the eleventh international conference on World Wide Web*, pp. 662–673. ACM Press, New York (2002)
4. Gal, A.: Managing uncertainty in schema matching with top-k schema mappings. *Journal of Data Semantics* 6, 90–114 (2006)
5. Gal, A., Modica, G., Jamil, H., Eyal, A.: Automatic ontology matching using application semantics. *AI Magazine* 26(1), 21–32 (2005)
6. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In: *Proceedings of the IEEE CS International Conference on Data Engineering*, pp. 117–140 (2002)
7. Melnik, S., Rahm, E., Bernstein, P.: Rondo: A programming platform for generic model management. In: *Proceedings of the ACM-SIGMOD conference on Management of Data (SIGMOD)*, San Diego, California, pp. 193–204. ACM Press, New York (2003)
8. Miller, R., Hernández, M., Haas, L., Yan, L.-L., Ho, C., Fagin, R., Popa, L.: The Clio project: Managing heterogeneity. *SIGMOD Record* 30(1), 78–83 (2001)
9. Rahm, E., Bernstein, P.: A survey of approaches to automatic schema matching. *VLDB Journal* 10(4), 334–350 (2001)
10. Sheth, A., Larson, J.: Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys* 22(3), 183–236 (1990)
11. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. *Journal of Data Semantics* 4, 146–171 (2005)

Enabling Flexible Execution of Business Processes

S. Jablonski, M. Faerber, F. Jochaud, M. Götz, and M. Igler

Chair for Databases and Information Systems, University of Bayreuth
Universitätsstrasse 30, 95447 Bayreuth, Germany
{Stefan.Jablonski,Matthias.Faerber,Florent.Jochaud,
Manuel.Goetz,Michael.Igler}@uni-bayreuth.de

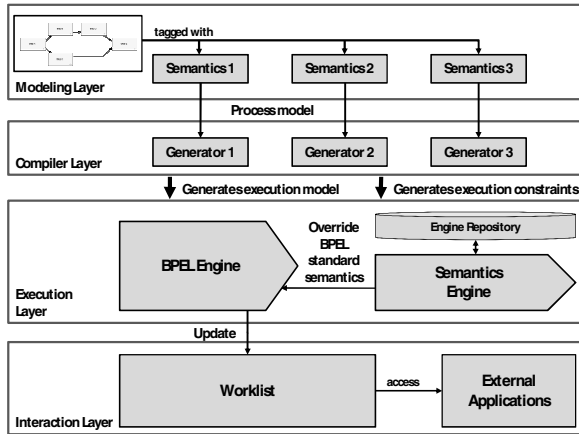
Abstract. Product development processes have been introduced and deployed in most engineering companies. Although these processes are generally well documented and understood it is still unclear how they can be supported by process management systems. In this paper we will propose a novel process management concept that better supports flexible cooperation between engineers in product development processes.

Modern product development is generally considered to be complex. It can be split into small work packages which are distributed among members of a development team; upon finalization the results need to be synchronized. Another characteristic of these scenarios is their iterative nature; also often the order of task execution is hard to prescribe. That is why business process management that is generally known for its wide support for cooperation in structured and repeatable processes [1] often fails in such iterative and flexible product development scenarios. FORFLOW¹, a joint research and pre-development project, aims at resolving this issue.

Although engineers can clearly define their target processes, typically real execution paths diverge from the recommended courses of actions. Thus engineers need a modeling and execution environment which allows specifying processes according to different execution semantics: from strict conventional workflow execution semantics – where the order of step execution is strictly prescribed – to flexible execution semantics – where a process management system only proposes steps which are recommended to execute next. The key issue in product development scenarios is to implement a process navigator (called Process Navigator) that is able to handle both strict and flexible execution semantics.

In the figure the concept of the Process Navigator is shown. Processes are modeled using the perspective oriented process modeling approach [2] which provides a decomposition of a process into orthogonal perspectives. The *Compiler Layer* transforms a POPM model into an executable process. It is composed of several generators one for each supported execution semantics. The functional aspects are converted into a BPEL process model (with BPEL4People extensions) whereas the control flow determining aspects are translated into a set of execution constraints which regulate the enactment of one specific execution semantics.

¹ The research work was partially funded by the Bayerische Forschungsförderung (research project "FORFLOW").



The BPEL model and the execution constraints are then interpreted in the *Execution Layer* which “enacts” the process. The *Semantics Engine* is evaluating execution constraints stored in the Engine Repository and is able to override the control flow defined in the BPEL model, enabling the flexible semantics. Users can select and start work steps from a worklist included in the *Interaction Layer*. After starting a step, the associated application is started and users can work on the process step.

Since the end of 2007 multiple project partners in the FORFLOW project are testing the process navigator successfully. This field test has demonstrated that the engineering domain could be greatly supported by a process management system that can perform processes according to multiple execution semantics.

Coordination of the work steps and the support for the cooperation of engineers in development teams, which have always been the strengths of process management systems, can only become apparent if users accept the systems. The flexible execution support that is provided by the process navigator is a key factor for this acceptance. The process management system and the provided execution semantics now reflect the typical iterative character of developing projects and no longer restrict users.

References

- [1] Georgakopoulos, D., Hornick, M., Sheth, A.: An overview of workflow management: from process modeling to workflow automation infrastructure. *Distributed and Parallel Databases* 3, 119–153 (1995)
- [2] Jablonski, S., Bussler, C.: *Workflow Management – Modeling Concepts, Architecture and Implementation*. International Thomson Computer Press, London (1996)

Collaborative Environment for Engineering Simulations with Integrated VR Visualization

Ismael H.F. Santos¹, Alberto B. Raposo², and Marcelo Gattass²

¹ CENPES, Petrobras Research Center, Ilha do Fundão,
21949-900, Rio de Janeiro, Brazil

² Tecgraf – Computer Graphics Technology Group, Department of Computer Science
PUC-Rio – Pontifical Catholic University of Rio de Janeiro, Brazil
ismaelh@petrobras.com.br,
{abraposo,mgattass}@tecgraf.puc-rio.br

Abstract. We present an SOA for executing engineering simulations and visualizing results in a Virtual Environment. Different technologies of group work are used to compose a Collaborative Problem Solving Environment that enables engineers to setup computations in an integrated environment.

Keywords: Scientific Workflows, Virtual Environments and SOA.

1 Collaborative Engineering Environment

In this work we present a Service-Oriented Architecture (SOA) for a Collaborative Engineering Environment (CEE) for assisting the control and execution of Petroleum Engineering projects. Those projects usually require the execution of a large number of engineering simulations, in our work encapsulated as engineering services, combined in different orders and rearranged in different subsets according to project requirements. By means of a Scientific Workflow Management System users are able to orchestrate the execution of simulations as workflow tasks, and as its last step, the most interesting cases can be selected for visualization in a collaborative session.

1.1 Riser Analysis Workflow

Floating production units (oil platforms) use ascending pipes, called risers, to bring the oil from the wellhead on the sea floor to the oil platform's separator system tanks (Fig. 1). To certificate the operation of the risers for their entire life cycle (30 years or so), simulations of the stress applied to the riser system are conducted based on extreme meteo-oceanographic conditions data (wind, tide and water currents). The riser analysis software used is Anflex [1], an internally developed Finite-Element-based structural analysis package.

For automating the process of validation and certification of riser analysis we have defined an Anflex-based riser analysis workflow controlled by the BPEL engine (Fig. 1). Web services were also created for taking care of the other parts of the workflow.

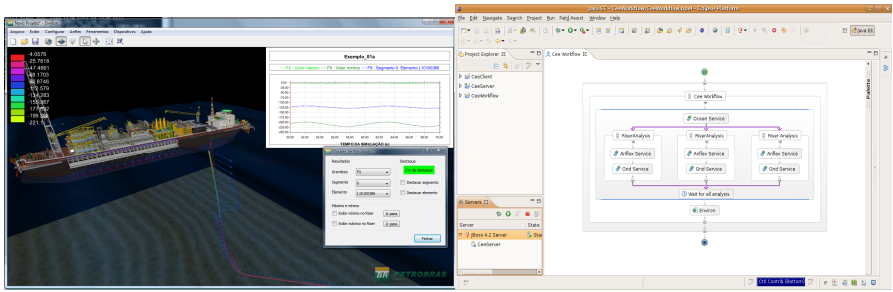


Fig. 1. Riser Analysis workflow

In the Collaborative Visualization Immersive Session, provided by Environ [2], results of the simulations can be analyzed by users in a desktop or in an immersive virtual environment. Among other resources, it is possible to playback the simulation, examine pipes, sea waves and ship movements, and track elements in the risers that are subjected to extreme conditions (e.g., high stress values). Annotations, private or public (shared) can also be created by the users, represented by distinct 3D-cursors, collaborating in a Environ Session where one of the users has created a private annotation that could be, for example, about an anomalous observed value (Fig. 2) .

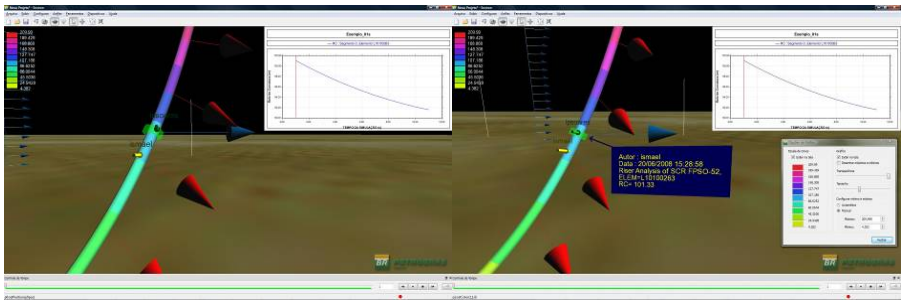


Fig. 2. Two users in a CEE collaborative visualization session

CEE is proving to be an effective Collaborative Problem Solving environment, allowing users to mitigate their problems during the execution of large and complex PE projects [3]. Although this work is focused on a solution for PE projects, we believe that the proposed CEE could also be used in other areas as well.

References

1. Mourelle, M.M., Gonzalez, E.C., Jacob, B.P.: ANFLEX - Computational System for Flexible and Rigid Riser Analysis. In: Proc 9th Intern. Symp. Offshore Engineering, Brazil (1995)
2. Raposo, A.B., Corseuil, E.T.L., Wagner, G.N., Santos, I.H.F., Gattass, M.: Towards the Use of CAD Models in VR Apps. In: ACM SIGGRAPH VRCIA, pp. 67–74 (2006)
3. Santos, I.H.F., Raposo, A.B., Gattass, M.: Finding Solutions for Effective Collaboration in a Heterogeneous Industrial Scenario. In: 7th CSCWD, pp. 74–79 (2002)

A Method for Searching Keyword-Lacking Files Based on Interfile Relationships

Tetsutaro Watanabe¹, Takashi Kobayashi², and Haruo Yokota¹

¹ Grad. School of Information Science and Engineering, Tokyo Institute of Technology, Japan
{tetsu@de,yokota}@cs.titech.ac.jp

² Grad. School of Information Science, Nagoya University, Japan
tkobaya@is.nagoya-u.ac.jp

Abstract. Traditional full-text searches cannot search keyword-lacking files, even if the files are related to the keywords. In this paper, we propose a method for searching keyword-lacking files named FRIDAL (File Retrieval by Interfile relationships Derived from Access Logs). The proposed method derives interfile relationship information from file access logs in the file server, based on the concept that those files opened by a user in a particular time period are related.

1 Introduction

Advances in information technologies have led to many types of multimedia data being stored as files in computer systems alongside conventional textual material. Moreover, the recent price drop for magnetic disk drives has accelerated the explosive increase in the number of files within typical file systems [1]. To find a desired file located at a deep node in the directory tree, several desktop search tools using full-text search techniques have been developed. However, their target is restricted to text-based files such as Office documents, PDFs, and emails. Other types of files, such as image files and data files, cannot be found by these full-text search tools because they lack search keywords. Even for text-based files, they cannot be found if they do not include directly related keywords. It becomes even harder if these files are located in different directories from the files that contain the keywords.

To address the demand for searching for these keyword-lacking files, we focus on the relationship between files that have been frequently accessed at about the same time. Although several researches for deriving interfile relationship from system call/OS event logs have been proposed [2,3], these methods need to modify OS of target systems and/or to install custom plugins and did not consider detail access patterns of target files.

In this paper, we propose a method for mining the file access logs in a file server to find interfile relationships and for searching keyword-lacking files that match with given keywords by using interfile relationships.

2 Proposed Method

First, we extract FUD (File Use Duration) of each files as the time between open-file and close-file from the file access logs. However, the actual duration of file use differs from the FUD because several applications do not keep file opened while using it and/or a user sometimes leaves his or her seat with files open.

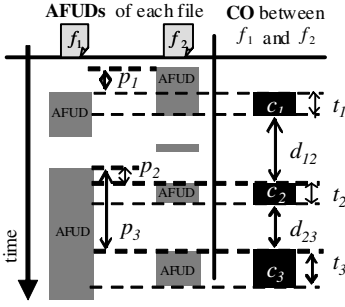


Fig. 1. Calculation of relationship elements

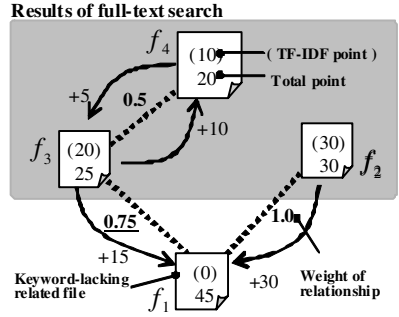


Fig. 2. Calculation of the point of files

In our proposed method, we prepare the *File Type List* to indicate which application keep file opened and calculate the *Active Time List* from the file access log. By using above two information, we extract *AFUD*(Approximate FUD) of each files.

We assume that strongly related files are used at the same time when executing the same task. To express this relationship, we introduce four “relationship elements”, where the term “CO” (co-occurrence) is defined as the overlap of two AFUDs; T (Total time of COs), C (Number of COs), D (Total time of the time span between COs) and P (Similarity of the timings of the open-file operations). Fig 1. shows how we calculate each elements. By using these four relationship elements, we define the weight of interfile relationship as follows: $R(f_i, f_j) = T^\alpha \cdot C^\beta \cdot D^\gamma \cdot P^\delta$. Finally we calculate file point of f_i by adding tf.idf point of f_j in proportion to the normalized $R(f_i, f_j)$ (Fig.2).

3 Conclusion

This paper presents a method for searching for files that lack keywords but do have an association with them. The proposed method derives interfile relationship by extracting AFUD of each files and calculating four “relationship elements” from AFUDs.

Although we cannot describe details due to limitations of space, we have implemented the proposed method FRIDAL as an experimental system. It can mines the interfile relationships from the access logs of Samba and performs the file point calculations by using interfile relationships and a full-text search engine, Hyper Estraier.

We also have evaluated its effectiveness by experiments. We have compared the search results for FRIDAL with a full-text search method, directory search method and a method used in Connections[2]. The experiment showed FRIDAL is superior to other methods in the 11-points precision and the recall/precision of the top 20.

References

1. Agrawal, N., Bolosky, W.J., Douceur, J.R., Lorch, J.R.: A Five-Year Study of File-System Metadata. *ACM Trans. on Storage* 3(3) (9) (2007)
2. Soules, C.A.N., Ganger, G.R.: Connections: using context to enhance file search. In: *Proc. SOSP 2005*, pp. 119–132 (2005)
3. Ohsawa, R., Takashio, K., Tokuda, H.: OreDesk: A Tool for Retrieving Data History Based on User Operations. In: *Proc. ISM 2006*, pp. 762–765 (2006)

Really Simple Security for P2P Dissemination of Really Simple Syndication*

Anwitaman Datta and Liu Xin

School of Computer Engineering, NTU Singapore
anwitaman@ntu.edu.sg, liu_xin@mail.ntu.edu.sg

1 Introduction

Using peer-to-peer overlays to notify users whenever a new update occurs is a promising approach to support web based publish subscribe systems like RSS. Such a peer-to-peer approach can scale well by reducing load at the source and also guarantee timeliness of notifications. However, malicious peers may stop propagating the updates or modify them, thus making the P2P mechanism useless or even harmful. We propose overlay independent randomized strategies to mitigate these ill-effects of malicious peers at a marginal overhead.¹

In the P2P approaches, generally a small subset of the end-users (*Rootpeers*) pull directly from the source and push any update downstream (to *Downstream peers*). Several such P2P approaches have been proposed [1], [2], [3]. Unlike other approaches, our approach puts the focus on the security issue, that is to protect the system against various malicious peers, without altering the behavior of currently deployed servers.

2 Our Approach

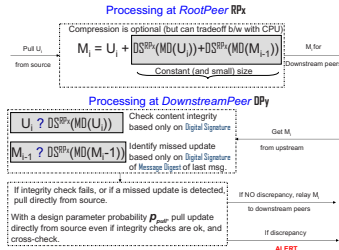
Disseminating content in a P2P manner has several risks. In the context of RSS feed dissemination, we identify three attack scenarios: (1) A malicious peer pushes downstream counterfeit content. (2) A malicious peer stops relaying contents. (3) A malicious peer provides false feedback about other peers (slanderer).

We propose randomized mechanisms to mitigate effect of all sorts of malicious peers by isolating the malicious peers using local information and actions at individual peers. Figure 1(a) summarizes the various processing steps at *Rootpeer* and *Downstream peer* to protect the system.

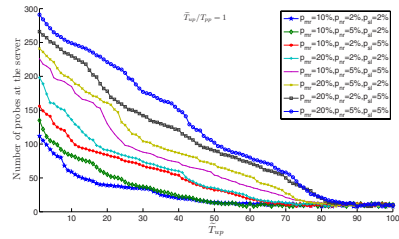
Rootpeers need to digitally sign the content they send downstream with their own private keys. Using message digest and digital signature, downstream peers can detect whether the update is modified or missed. If the *Rootpeer* itself is malicious, it can be detected by downstream peers by checking the server randomly (with a probability P_{pull}). To handle the false alert, we propose K-probe.

* Part of the research presented in this paper has been supported by A*Star SERC Grant No: 0721340055.

¹ Please refer to the longer version of the paper for more details at <http://www3.ntu.edu.sg/home/Anwitaman/>



(a) $MD(\cdot)$ is the operation to compute message digest. $DS^i(\cdot)$ is the operation to compute the digital signature using peer i 's private key. The verification operation '?' is conducted using the corresponding pubic key.



(b) Number of probes at the server for different fractions of all kinds of malicious peers and for $\tilde{T}_{up}/T_{PP} = 1$, $K=4$.

Fig. 1.

When a peer receives an alert, it probes K random peers to verify it. If all the responses support this alert, then this peer considers it to be genuine. If the responses are not identical, this peer has to resort to probing the server directly. Rigorous evaluations show that our approach eliminates all malicious peers at low overhead and even before the elimination of misbehaving peers is complete, performs robustly against attacks.

3 Results

We study the effectiveness of our defense mechanisms under diverse attack scenarios. We denote that each peer probes the server every T_{PP} time with a probability. The average time between two updates is \tilde{T}_{up} . To verify the alert, a peer will probes K other random peers. From Figure 1(b), we can see that higher percentage of malicious peers delay the convergence time because it will take more time to detect the malicious peers, thus increasing the load at the server. Nevertheless, the number of probes at the server again and always converges, which corresponds to isolating all malicious peers from the dissemination network.

References

1. Datta, A., Stoica, I., Franklin, M.: LagOver: Latency Gradated Overlays. In: ICDCS (2007)
2. Ramasubramanian, V., Peterson, R., Sifer, E.G.: Corona: A High Performance Publish-Subscribe System for the World Wide Web. In: NSDI (2006)
3. Sandler, D., Mislove, A., Post, A., Druschel, P.: FeedTree: Sharing Web micronews with peer-to-peer event notification. In: Castro, M., van Renesse, R. (eds.) IPTPS 2005. LNCS, vol. 3640, Springer, Heidelberg (2005)

Mining and Analyzing Organizational Social Networks Using Minimum Spanning Tree

Victor Ströele A. Menezes, Ricardo Tadeu da Silva, Moisés Ferreira de Souza,
Jonice Oliveira, Carlos E.R. de Mello, Jano Moreira de Souza,
and Geraldo Zimbrão

COPPE/UFRJ – Graduate School of Computer Science, Federal University of Rio de Janeiro.
PO Box 68.513, 21945-970 - Rio de Janeiro, RJ, Brazil - +55 21 2562.8696
{stroele,rick,moises,jonice,carlosmello,jano,
zimbrao}@cos.ufrj.br

Abstract. This work focuses on using data mining techniques to identify intra and inter organization groups of people with similar profiles and that could have relationships among them. Our clustering method identifies clusters with a link mining-based technique that uses the minimum spanning tree to construct group hierarchies. In this paper we analyze the scientific scenario in Computing Science in Brazil, assessing how researchers in the best universities and research centers collaborate and relate to each other.

Keywords: Data mining, scientific social networks, Social networks analysis.

A social network reflects a social structure which can be represented by nodes (individuals or organizations) and their relations. Relations can be specific types of interdependency (such as idea) or more specific relationships (as financial exchanges, friendship, communication, and others). Several efforts have been made in order to analyze social networks [1] [2]. From the data mining standpoint, a social network is a heterogeneous and multirelational data set represented by a graph [3].

The purpose of this work is to group people with common characteristics and relationships in the social network. We use this approach to study the scientific social network in Brazil, in the Computing Science scenario.

Scientific social networks are social networks where two scientists are considered connected if they have co-authored a paper [4]. The nodes of the graph are represented by researchers and the edges are relationships between each pair of researchers. These relationships may be: Project Participation; Co-authored publications; Advising work; and other types of scientific production. In addition to relationships, each of the professors has their individual profile, such as: Academic Training; Research and activity area; Number of Journal Publications; and others.

Professors are linked to each other either directly or indirectly. This association may be stronger or weaker according to the degree of relationship between them. In order to identify groups of people, our method follows a strategy based on pruning edges of the social graph. Figure 1 shows results generated by our methodology. The largest regions illustrate the Brazilian institutions, and the smaller circles with the same number illustrate the groups within an organization. The edges represent only the strongest relationships (Minimum Spanning Tree, PRIM [5]).

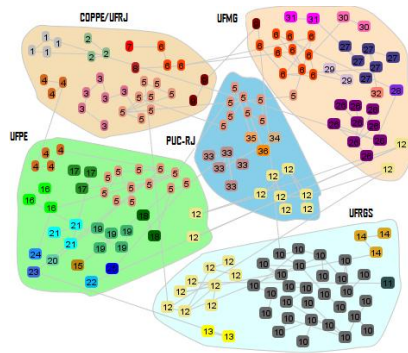


Fig. 1. Inter and Intra institutional relationship

Analysing the results, we concluded that there are few professors who have the profile to work more closely with another professor from another university. Thus, the relationship between the institutions is linked by some professors with strong relationships. This fact can be seen through the relationships between UFRJ and UFMG. On the other hand, there are universities where their external relationships are formed by a large number of researchers. Therefore, this type of institution has several professors with weak external relationships. This case can be seen in UFRGS.

The degree of relationship between professors was evaluated considering the social network in a more detailed way. This analysis was based on internal and external relationships, from the point of view of each researcher. As it was expected, internal relationships are generally stronger than external ones. Thus, it was found that professors of the same institution have a greater tendency to publish together than professors from different institutions.

All data used in the experiments described above are real. Thus, we have guarantees on the validity of the structure of the social network that was studied. The results were validated by an interview with the professors of one of the universities.

We can conclude that the results obtained enabled us to identify several features of the scientific social network. With the analysis of this social network, it was possible to determine the degree of relationship between the educational institutions.

References

1. Wasserman, S., Faust, K.: *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge (1994)
2. Freeman, L.: Centrality in social networks: Conceptual clarifications. *Social Networks* 1, 215–239 (1979)
3. Han, J., Kamber, M.: *Data Mining: Concepts and techniques*, 2nd edn. Morgan Kaufmann Publishers, USA (2006)
4. Newman, M.E.J.: The structure of scientific collaboration networks. In: *Proceedings of the National Academy of Science USA* 98, 404–409 (2001)
5. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, E.C.: *Introduction to Algorithms*, 2nd edn. MIT Press and McGraw-Hill (2001)

MADIK: A Collaborative Multi-agent ToolKit to Computer Forensics

Bruno W.P. Hoelz, Célia G. Ralha, Rajiv Geeverghese, and Hugo C. Junior

Computer Science Department
University of Brasília
Campus Universitário Darcy Ribeiro
Caixa Postal 4466 - CEP 70.919-970 - Brazil
werneck.bwph@dpf.gov.br, ghedini@cic.unb.br,
vectorius@yahoo.com.br, hugo.csj@gmail.com

Abstract. In this article, we present **MADIK**, a **Multi-Agent Digital Investigation ToolKit** to help experts during the forensic examination process. MADIK uses a four layer multi-agent architecture, as a metaphor to the organizational hierarchy levels: strategic, tactical, operational and specialist. The proposed architecture and tool was developed under a blackboard approach, implemented with *Java Agent DEvelopment Framework - JADE*, using *Java Expert System Shell - JESS* as an inference engine. We have done some experiments with MADIK using real data, on stand alone and distributed environments with encouraging results.

Keywords: computer forensics, collaborative multi-agent systems, digital investigation, JADE, JESS.

1 The Proposed Approach and Results

Computer Forensics consists of examination and analysis of computational systems, which demands a lot of resources due to the large amount of data involved. Frequently, at real computer forensic cases, experts can't define at first what evidence is more relevant to the incident or crime under investigation. Thus, a pre-analysis of the suspect machines would limit the number of evidences collected for examination, reducing the time of investigation by the forensic experts.

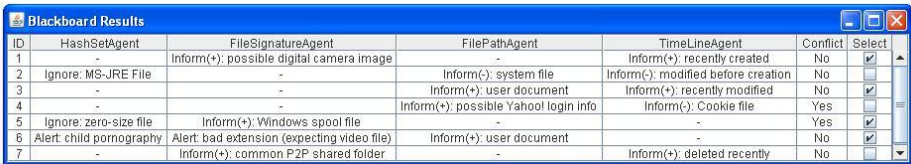
Forensic examination of computer systems consists of several steps to preserve, collect and analyze evidences found in digital storage media, so they can be presented and used as evidence of unlawful actions [1]. But there is a lack of intelligent and flexible tools to help forensic experts with the pre-analysis phase, and also with a concrete cross-analysis of large number of potential correlated evidences. What really happens is that the computers and storage media are analyzed separately and a large number of potentially related evidences are lost during the examinations.

This research uses a multi-agent approach which has been adequate to tackle these problems, specially regarding the cooperative action of the autonomous specialized agents [2]. Thus, we have defined different specialized intelligent

agents that act based on the experts knowledge of the technical domain. The specialized intelligent software agents already implemented include: (i) HashSetAgent, similar to the KFF[®] functionality found on AccessData Forensic ToolKit (FTK[™]), (ii) FileSignatureAgent; (iii) FilePathAgent and (iv) TimelineAgent. The last three agents are similar to some functionalities found on Guidance EnCase[™].

Figure 1 presents a sample of the blackboard results using the four specialized agents described. Each column contains the recommendation and the description from the agents, while each line corresponds to a specific file being examined. Notice that not every file has a recommendation from all agents, but the more specialists we have, the broader coverage we expect to achieve.

Also there are files with diverging recommendations from different agents, e.g. ID 5 has Ignore:zero-size file and Inform(+):Windows spool file. These conflicts are presented in the *Conflict* column. The select box mark the operational manager conflict resolution results. At the moment, the operational manager currently uses a naïve approach, that considers the presence of a positive bias more important than a negative one. This results in some excessive inclusion of files and represents additional work to the human reviewer, but if it would consider the negative bias first, some important evidence could be left behind.



ID	HashSetAgent	FileSignatureAgent	FilePathAgent	TimeLineAgent	Conflict	Select
1	-	Inform(+): possible digital camera image	-	Inform(+): recently created	No	<input checked="" type="checkbox"/>
2	Ignore: MS-JRE File	-	Inform(-): system file	Inform(-): modified before creation	No	<input type="checkbox"/>
3	-	-	Inform(+): user document	Inform(+): recently modified	No	<input checked="" type="checkbox"/>
4	-	-	Inform(+): possible Yahoo! login info	Inform(-): Cookie file	Yes	<input type="checkbox"/>
5	Ignore: zero-size file	Inform(+): Windows spool file	-	-	Yes	<input checked="" type="checkbox"/>
6	Alert: child pornography	Alert: bad extension (expecting video file)	Inform(+): user document	-	No	<input checked="" type="checkbox"/>
7	-	Inform(+): common P2P shared folder	-	Inform(+): deleted recently	No	<input type="checkbox"/>

Fig. 1. Sample of blackboard results

This paper described a four layer multi-agent architecture, as a metaphor to the organizational hierarchy levels. A proof-of-concept prototype was implemented called MADIK, which has been tested with real data. MADIK seeks to perform forensic examinations in a more intelligent and flexible way, since the toolkit includes distribution of tasks in the examination processes and allows for correlation of findings. The results obtained with MADIK were compared to those obtained by a human examiner. Although not evaluated numerically, we noticed that the results were similar and encouraging.

References

- [1] Beebe, N., Clark, J.G.: A hierarchical, objectives-based framework for the digital investigations process. *Digital Investigation* 2(2), 147–167 (2005)
- [2] Wooldridge, M.: *An Introduction to MultiAgent Systems*. John Wiley & Sons, Ltd., Sussex (2002)

Modularizing Monitoring Rules in Business Processes Models

Oscar González^{1,2,*}, Rubby Casallas², and Dirk Deridder¹

¹ Vrije Universiteit Brussel, SSEL Lab, Pleinlaan 2, 1050 Brussel, Belgium

² University of Los Andes, TICSw Group, Carrera 1 N° 18A 10, Bogotá D.C., Colombia
{o-gonzal, rcasalla}@uniandes.edu.co, dirk.deridder@vub.ac.be

1 Context

Business process management systems contain *monitoring, measurement and control (MMC)* specifications to enable the identification of problems and solutions to improve business processes. Business processes facilitate the integration of human and technological resources in an organization, according to a set of activities that fulfil a policy goal [1].

Currently, several business-process monitoring, measurement and control solutions are available [2]. However, MMC specifications typically are *implicitly* encoded in the low-level implementation of the workflow system. This results in tangled and scattered MMC knowledge in the underlying process code. It is clear that this decreases the maintainability and reusability of the MMC specifications since they are not specified in a modular fashion. Furthermore, specific knowledge about the overall system implementation is required if the MMC requirements evolve. Due to the entanglement with the low-level implementation, this requires a level of expertise that is normally available to technical developers. This is unfortunate since business experts typically express the MMC requirements at a high level and in terms of the business domain (as opposed to the technical implementation). The task becomes even more complicated because the MMC specifications involve data that is not readily available in one location of the process code. In addition the majority of existing approaches focus on describing the MMC specification in terms of the process execution instead of the data flow of the process. Consequently, when existing MMC specifications need to be adapted, the different pieces of process code must be manually localized after which adaptations will occur at several places.

2 Monitoring Approach

In order to overcome these problems we created a domain specific language (DSL) to capture MMC specifications according to the specific domain of the process that is being modelled (A). This provides the mechanisms to express MMC specifications at the process definition level (e.g., in BPMN) instead of at the low-level implementation. In this specification it is also possible to refer explicitly to the data

* Supported by the Flemish Interuniversity Council (VLIR) funded CAMELOS project and the “Instituto Colombiano para el Desarrollo de la Ciencia y la Tecnología COLCIENCIAS”.

flow of the process (B). We also provide the necessary mechanisms to integrate the high-level MMC specifications with the existing process models (C).

A) High-Level of Abstraction. Our DSL enables the domain experts to express the MMC specifications in terms of domain concepts that they already know, offering a high-level specification. The MMC specifications that are described at the domain level can be automatically translated to the target language, which makes it independent of a particular implementation technology. The measurement part of the specification describes the new monitoring data (monitoring concepts) that must be created from the existing monitoring concepts or from data originally defined in the process. The monitoring part of the specification describes what data must be recovered during process execution and when it should occur. The control part of the specification describes the control actions to take according to the information recovered from monitoring specification when a condition is satisfied.

B) Semantic Specification. Although we consider the behavior of the control flow, we also consider the data entities in the process to be the main elements to be analyzed. Therefore we provide support to enrich the initial data of the process with new monitoring data (expressed as domain concepts). This monitoring data can then be used as building blocks in the high-level language, enriching the measurement capabilities towards a semantic specification. Currently we focus a large part of our research on the association of a data model to the process model. With this data model we expect to raise our capabilities to express the MMC specifications in terms of (business) domain concepts.

C) Domain Composition. Two approaches have been considered to compose the MMC specification with the process model. Both at the domain level and at the implementation level, we consider the integration of MMC specifications with the base process control-flow using an aspect-oriented approach [3]. This approach provides the encapsulation of MMC specifications, which can be reused to add new functionalities with slight modifications. At the domain level, part of the MMC specification is transformed into an independent process model, thus if an error occurs in the monitoring process, the normal flow can continue without interruption. In addition, the aspect interferences and interactions can be validated and resolved before the process is in execution. Currently we are working in the transformation and composition at the implementation level of the information in the MMC specification that cannot be represented in the high-level process model.

References

1. Van der Aalst, W.M.P., Ter Hofstede, A.H.M., Weske, M.: BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
2. Miers, D., Harmon, P., Hall, C.: The 2007 BPM Suites Report – Version 2.1. Business Process Trends (BPTrends), http://www.bptrends.com/reports_toc_01.cfm
3. Jacobson, I.y., Ng, P.-W.: Aspect-Oriented software development with use cases. Addison-Wesley, United States (2005)

An Optimal Approach for Workflow Staff Assignment Based on Hidden Markov Models^{*}

Hedong Yang¹, Chaokun Wang², Yingbo Liu¹, and Jianmin Wang²

¹ Department of Computer Science and Technology, Tsinghua University
Beijing, China, 10084

{yanghd06, lyb01}@mails.tsinghua.edu.cn

² Tsinghua National Laboratory for Information Science and Technology (TNList)
School of Software, Tsinghua University, Beijing, China, 10084

{chaokun, jimwang}@tsinghua.edu.cn

Abstract. Staff assignment of workflow is often performed manually and empirically. In this paper we propose an optimal approach named SAHMM (Staff Assignment based on Hidden Markov Models) to allocate the most proficient set of employees for a whole business process based on workflow event logs. The Hidden Markov Model(HMM) is used to describe the complicated relationships among employees which are ignored by previous approaches. The validity of the approach is confirmed by experiments on real data.

Keywords: Workflow, staff assignment, Hidden Markov Model.

1 Introduction

Workflow staff assignment, namely to allocate right persons for right tasks at right time, is a challenging problem when a workflow contains dozens of tasks. Human beings play an important role in many business processes because they cannot be replaced thoroughly by machines. Hence successful staff assignment becomes the basis of a successful business process. Traditionally staff assignment is often performed by allocators according to personal experiences and/or empirical rules, which requires allocators to know requirements of all tasks and qualifications of all employees for whole business processes. It is really a hard job for allocators when a workflow is complicated. Business reformation and business processes improvement would make this even harder.

Solutions available for this problem seldom consider the dependant relationships among employees. Traditional algorithms for resource allocation can be used for staff assignment if only the qualifications of employees were considered. Recently computer-aided staff assignment has been studied based on workflow event logs, including mining rules [1,2] and proposing candidate employees [3].

* This research was partially supported by the National Key Basic Research Program of China (No. 2002CB312006, No. 2007CB310802), the National High Technology Research and Development Program of China (No. 2008AA042301) and the National Natural Science Foundation of China (No. 90718010).

Although these researches lead to a new direction, none of them allocates employees for a whole business process which requires employees to work cooperatively.

2 Staff Assignment Based on Hidden Markov Models

Definition 1. *Proficiency-oriented Log-based Staff Assignment:* Given a workflow model $WF = (P, T, F, E)$ and a workflow event log L , the objective is to find an employee set $R = \{R_1, R_2, \dots, R_n\}$ based on L which has the highest proficiency for WF , where $T = \{T_1, T_2, \dots, T_n\}$, $E = \{E_1, E_2, \dots, E_n\}$, $R_i \in E_i$.

The key point of the proposed approach, SAHMM, is to model such a problem as a decoding problem of the HMM [4] where tasks (T) are observable states and candidate employees (E) hidden states. The proficiency of an employee set is divided into the *individual proficiency* of employees for specific tasks and the *tacit degree* among all employees which would be described with probability functions $\mathbf{P}(T_i|E_i)$ and $\mathbf{P}(E_{i+1}|E_i)$ respectively. The two functions extracted from L and an uniform distribution on E forms the parameters for the HMM. The structure of the HMM is transformed from WF . A virtual node V would be added to connect two branches having the same split and join nodes where $\mathbf{P}(V|E_V) = \mathbf{1}$, $\mathbf{P}(E_N|E_V) = \mathbf{1}$ and $E_V = E_N$. N is the next node to V . We proved adding such virtual nodes will not affect calculating the optimal result.

Experiments results show SAHMM works well. We carried out experiments on real log data from two vehicle manufacturing companies. The data covers 6569 successful executions of 44 business processes and 610 tasks performed by 326 employees. Experiment results show that 95.69% of employees has a preferred consecutive workmate which confirms the existence of the dependant relationships among employees. Each workflow had been mined out an optimal employee set which testified the validity of SAHMM.

3 Conclusion and Future Work

We discussed an optimal approach, SAHMM, which offers a novel way to allocate employees for a whole business process. A Hidden Markov model is adopted as a tool to describe dependant relationships among employees working cooperatively. We proposed methods to build the structure of the model from a workflow and to calculate its parameters based on workflow event log data. Experiment results confirmed the existence of such relationships and the validity of SAHMM. The approach considers little of work load which will be improved in the future.

References

1. Ly, T., Rinderle, S., Dadam, P., Reichert, M.: Mining staff assignment rules from event-based data (2005)
2. Rinderle-Ma, S., van der Aalst, W.: Life-cycle support for staff assignment rules in process-aware information systems. BETA Working Paper Series, WP 213, Eindhoven University of Technology (2007)

3. Liu, Y., Wang, J., Yang, Y., Sun, J.: A semi-automatic approach for workflow staff assignment. *Computers in Industr.* (2008) (in Press, Corrected Proof)
4. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2), 257–286 (1989)

Behavioral Compatibility of Web Services

Zhangbing Zhou, Sami Bhiri, Walid Gaaloul, Lei Shu, and Manfred Hauswirth

Digital Enterprise Research Institute, National University of Ireland at Galway
firstname.lastname@deri.org

Abstract. Current methods for compatibility analysis consider only direct service interactions where no mismatches are allowed. However mismatches usually exist among Web services, and their interactions are often carried out with the help of mediators. In addition, current approaches don't give precise evaluation of partial compatibility which is useful for ranking and selecting candidate services.

In this paper, we consider compatibility beyond direct service interaction where mediation can be applied. We present an approach to check if two business processes can be mediated or not. Our approach also enables better evaluation of compatibility by quantifying the degree of compatibility as a number between 0 and 1.

Keywords: Compatibility, Business Process, Mediated Web Service Interaction, Process Scenario, View.

1 Introduction

A main promise of Web service architecture is to support seamless service interactions of diverse business processes encapsulated as Web services [3]. To avoid runtime errors, business processes are required to agree on certain constraints, i.e. behavioral compatibility [1]. However, in Web service domain, it is difficult, if not impossible, to find two business processes that are completely compatible. An interaction is usually carried out with the help of data and/or process mediators [2], so-called mediated service interactions. In this paper we consider an extended vision of compatibility that goes beyond direct services interactions and takes into account possible mediations.

Nowadays, most approaches give a binary answer, which is not very helpful because business processes often can interact in some, if not all, cases. [1] provided a ternary answer for compatibility. Even this is more precise, it does not help much to rank and select service candidates especially for those partially compatible. Our approach gives more precise answer on how far two processes are compatible by returning a number between 0 and 1. This number quantifies the degree of compatibility between them.

2 Generating a View for a Business Process Scenario

Only a part of a business process is involved in a particular interaction depending on the guard function status of its *Switch* control elements. A scenario of business process p is a sub-process of p that may be enacted in a particular interaction.

A view is a virtualisation of a scenario where activities, which have neither control nor mandatory data dependencies between them, are folded together in what we call virtual activities. A view is generated from a scenario using reduction rules which aim to replace a *Sequence*, *Flow*, or *While* block by one or a sequence of virtual activities.

3 Computing the Degree of Compatibility

For two scenarios, if messages exchanged between them can carry out a successful interaction, the messages can lead their views from their initial virtual activities to their final virtual activities. Then these two views are called *compatible*.

Based on pairwise compatibility of their views, we can define the degree of compatibility for two public processes p_1 and p_2 . We assume that there are n_1 views in p_1 . For a view v_i ($1 \leq i \leq n_1$) in p_1 , we define a function $comp(v_i | p_2)$ to specify whether there is a compatible view in p_2 if $comp(v_i | p_2) = 1$, or $comp(v_i | p_2) = 0$ otherwise. Thus, the degree of compatibility for p_1 to p_2 is:

$$Compatibility(p_1, p_2) = \frac{\sum_1^{n_1} comp(v_i | p_2)}{n_1} \quad (1)$$

Compatibility at a view level is a *symmetric* relation. However, compatibility at a public process level is an *antisymmetric* relation. Below we define three classes of compatibility for two public processes p_1 and p_2 :

- *No compatibility* if $Compatibility(p_1, p_2) = 0$, which means that two public processes cannot interact in any case.
- *Partial compatibility* if $0 < Compatibility(p_1, p_2) < 1$, which means that one public process can interact with another in any case
- *Full compatibility* if $Compatibility(p_1, p_2) = 1$, which means that one public process can interact with another in at least one but not all cases.

4 Conclusion

We have identified that current methods for checking compatibility are limited to support service interactions for Web service based business processes. We have proposed our approach considers the compatibility as if business processes can be mediated. In the future, we will consider the “typical behaviors” by assigning different importance weight to different scenarios to improve our work.

References

1. Benatallah, B., Casati, F., Toumani, F.: Representing, analysing and managing web service protocols. *Data & Knowledge Eng.* 58(3), 327–357 (2006)
2. Fensel, D., Bussler, C.: The Web Service Modeling Framework WSMF. *Journal of Electronic Commerce Research and Applications*, 113–137 (2002)
3. Yu, Q., Liu, X., Bouguettaya, A., Medjahed, B.: Deploying and managing Web services: issues, solutions, and directions. *The VLDB Journal* 17(3), 537–572 (2008)

A Reverse Order-Based QoS Constraint Correction Approach for Optimizing Execution Path for Service Composition*

Kaijun Ren^{1,2}, Nong Xiao¹, Jinjun Chen², and Junqiang Song¹

¹ College of Computer, National University of Defense Technology, Changsha, Hunan 410073, P.R. China

² CS3-Centre for Complex Software Systems and Services, Swinburne University of Technology, Melbourne 3122, Australia
{Renkaijun, Junqiang}@nudt.edu.cn, xiao-n@vip.sina.com, jchen@swin.edu.au

In service oriented computing systems, a business process can be exposed as a composite service which consists of a set of logically connected sub-services. For each service in the composition, many service providers can offer the same function but may different QoS. In the general service composition, when a user submits a request, overall QoS constraints called end-to-end QoS composition's requirements, for example, time should be less than one hour, and cost should be less than 60\$, can be transmitted at the same time. As such, how to effectively coordinate individual QoS constraints for single service to achieve the best overall QoS benefits without violating such end-to-end QoS constraint requirements has been a critical issue. With an increasing number of abstract services in a service composition, the possibility of execution path by selecting different service providers for each abstract service blows up exponentially. Therefore, service selection problem for service composition is a computational-hard problem, which can be regarded as a Multiple choice Multiple dimension Knapsack Problem (MMKP) that has been proved np-hard [1, 2, 3]. Recently, a lot of approaches such as graph-based techniques[4], runtime adaptation-based techniques[5], Service Level Agreement(SLA), negotiation and auction based techniques[6], Integer Linear Programming (ILP) based techniques[7] have been proposed to resolve overall QoS constraints for optimizing execution path in a service composition. No matter what the merits and the importance current existing methods have, they rely on directly judging constraint conditions to detect multiple paths for picking out a critical execution path, which easily produces a high-time complexity and even an unsatisfactory result in comparison to the best path. As such, the issue on resolving overall QoS constraints to achieve an optimal execution path has not yet been well addressed.

Motivated by the aforementioned issue, we propose a Reverse Order-based(which distinguishes the traditional order) QoS constraint correction approach for resolving end-to-end QoS constraint requirements. Instead of directly relying on judging constraint conditions to detect multiple paths for picking out an optimized one, our

* This paper is supported by Swinburne Dean's Collaborative Grants Scheme 2007-2008, and by Swinburne Research Development Scheme 2008, and by the National "973" Research Plan Foundation of China under Grant No. 2003CB317008.

methods first build an initial optimal execution path by employing the local optimization policy without considering user-expressed end-to-end QoS composition constraints. Based on the initial path, the global QoS computing models and violation checking models can detect all occurred constraint violations. For each violation, a Reverse Order-based correction algorithm is proposed to recursively correct such violations by reselecting critical service providers. Finally, an optimized execution path can be rebuilt to meet overall end-to-end QoS composition requirements.

Basing on the above-mentioned methods, we have finished the basic experiment test. In the experiment, we use a service test collection from QSQL previously built in [8] where 3500 services have been included to generate service composition flows. Then, candidate service providers are produced as a random value for being associated with corresponding abstract services. Particularly, different QoS attributes such as price, time, and reliability which are randomly generated with a uniform distribution, are assigned with each service provider. As a result, the preliminary experiment shows that our methods can make the time-complexity of service composition decrease significantly by avoiding detecting multiple unnecessary paths. Simultaneously, a near-optimal execution path can be achieved to meet all end-to-end QoS composition requirements. Our future work is to combine the Reverse Order-based method with our existing relative workflow framework to develop a concrete business process application.

References

1. Ardagna, D., Pernici, B.: Adaptive Service Composition in Flexible Processes. *IEEE Transaction on Software Engineering* 33(6), 369–383 (2007)
2. Yu, T., Zhang, Y., et al.: Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints. *ACM Transactions on the Web*, Article 6 1(1), 6:1–6:26 (2007)
3. Gu, X., Nahrstedt, K.: On Composing Stream Applications in Peer-to-Peer Environments. *IEEE Transactions on Parallel and Distributed Systems* 17(8), 824–837 (2006)
4. Gekas, J., Fasli, M.: Automatic Web Service Composition Based on Graph Network Analysis Metrics. In: *The Proceeding of the 5th International Conference on ontologies, Databases and Applications of Semantics*, Springer, Agia Napa, Cyprus (2005)
5. Canfora, G., Penta, M., et al.: QoS-Aware Replanning of Composite Web Services. In: *The Proceeding of 2005 International Conference on Web Service*, Orlando, USA (July 2005)
6. Yan, J., Kowalczyk, R., et al.: Autonomous service level agreement negotiation for service composition provision. *Future Generation Computer Systems-the International Journal of Grid Computing Theory Methods and Applications* 23(6), 748–759 (2007)
7. Zeng, L., Benatallah, B., et al.: QoS-Aware Middleware for Web Services Composition. *IEEE Transaction on Software Engineering* 30(5), 311–327 (2004)
8. Ren, K., Liu, X., et al.: A QSQL-based Efficient Planning Algorithm for Fully-automated Service Composition in Dynamic Service Environments. In: *The Proceeding of 2008 IEEE International Conference on Services Computing (SCC 2008)*, IEEE Computer Society Press, Honolulu, Hawaii, USA (2008)

Data Mining of Specific-Domain Ontology Components

J.R.G. Pulido¹, M.A. Aréchiga², and M.E.C. Espinosa³

¹ Faculty of Telematics, University of Colima, México
jrgp@ucol.mx

² Faculty of Telematics, University of Colima, México
mandrad@ucol.mx

³ Info Systems and Comp Dept, University of Valencia, Spain
mcabello@dsic.upv.es

Abstract. This paper describes an approach for eliciting ontology components by using knowledge maps. The knowledge contained in a particular domain, any kind of text digital archive, is portrayed by assembling and displaying its ontology components.

1 Introduction

The present-day web and its always growing amount of web pages must be broken into smaller pieces that slowly but surely will be transformed into semantic web [1] pieces. Reaching specific web pages is a gigantic challenge having into account that current search engines only contain a small percentage of the total of documents in the web. In other words, these documents can be read, by a web browser for instance, but not understood.

2 Related Work

Knowledge can be formalized by means of five kind of components: **Concepts** In an academic domain, some concepts that we find are: *University, Research group, researcher.* **Relations** In an academic domain, some relations that we find are: *works for, collaborates with, has project.* **Functions** Some functions that we find in an academic domain are: *leader of project, head of school, lab officer.* **Axioms** Some axioms that we may find in an academic domain are: *CONACyT funds mexican research projects, CGIC funds University of Colima research projects.* **Instances** Some instances that we may find in an academic domain are: *JRG Pulido, University of Colima, FRABA 498-07.*

Basically, *facts* represent relations that hold between instances, and *claims* represent assertions that are true for instances. Reasoners may use these components for inferring new data from knowledge bases.

3 Methods

Self-Organizing Maps (SOM) is a learning approach that can be viewed as a model of unsupervised learning and an adaptive knowledge representation scheme [2]. After this

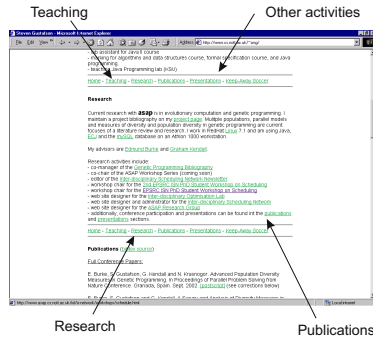


Fig. 1. People include research interests, teaching duties, and so forth in their pages

stage, some other ontology tools such as editors can be used to organize this knowledge. Then, it can be embedded into the digital archive where it was extracted from by means of any of the ontology languages that exist [3].

4 What We Are Looking for

For our analysis we have only retrieved local hyperlinks from an academic domain. Each web page contains information related to its owner, for instance, *research* interests, *teaching* duties, *projects* on which they are working, and so on (fig 1).

A University consists of a number of entities, for instance *school* and *person*. One School has generally more that one *research group*. A person usually plays more that one *role* within the school, *member* of a research group for instance, and has a number of *publications*. A specific domain such an academic domain requires knowledge from the experts in order to produce a complex ontology.

5 Conclusions

The present-day web in this way can be broken into smaller pieces that slowly but surely will be transformed into semantic web pieces. Should that be done manually, then the semantic web will not become a really in the next couple of decades due to this bottleneck. Ontology learning tools are essential for the realization of the semantic web for the job to be done is quite complex [5].

References

- [1] Berners-Lee, T., et al.: The Semantic Web. Scientific American 284(5), 34–43 (2001)
- [2] Kohonen, T.: Self-Organizing Maps, 3rd edn. Information Sciences Series. Springer, Berlin (2001)
- [3] Pulido, J.R.G., et al.: Ontology languages for the semantic web: A never completely updated review. Knowledge-Based Systems 19(7), 489–497 (2006)

- [4] Pulido, J.R.G., et al.: In the quest of specific-domain ontology components for the semantic web. In: Ritter, H., et al. (eds.) 6th Intl. Workshop on Self-Organizing Maps (WSOM 2007), Neuroinformatics group, Bielefeld University, Germany, pp. 1–7 (2007) CD edition
- [5] Pulido, J.R.G., et al.: Artificial learning approaches for the next generation web: part II. In: Ingeniería Investigación y Tecnología, UNAM (CONACyT), México, 9(2), 161–170 (2008)

Distributed Data Mining by Means of SQL Enhancement

Marcin Gorawski and Ewa Pluciennik

Institute of Computer Science, Silesian University of Technology, Akademicka str. 16,
44-100 Gliwice, Poland
{Marcin.Gorawski, Ewa.Pluciennik}@polsl.pl

1 Introduction

An analysis of a huge amount of information is feasible only if information systems are used. First, information needs to be accumulated and stored in a persistent structure enabling effective data access and management. The main aspects of nowadays data processing are: storing data in (mostly relational) databases, improving data processing efficiency by parallel analysis [1], distributed processing (necessary for institution consisting of autonomous, geographically distributed departments), query languages (SQL) remain a fundamental way to access data in databases, data analysis often includes data mining (building data models describing data characteristics or predicting some features) [2].

Regarding the above mentioned circumstances authors propose an enhancement of SQL for data mining of a distributed data structure. Basic assumption is a complete, horizontal data fragmentation and an explicit model format. Building global data model consists of two stages. In the first one, local models are built in a parallel manner. Second one consists of combining these models into a global data picture. Detailed description of combining methods regarding global classification models authors presented in [3].

2 EwSQL (Explore with SQL)

SQL enhancements enabling data mining are still not standardized. In 2001 the SQL/MM standard has emerged [4]. This standard defines issues concerning data mining using object features of databases (defining object types and methods) and XML as a data structure definition language. The standard does not define any new SQL constructions for data mining. Such constructions arise within the confines of scientific researches, for example DMQL [5], MSQL [6], Mine Rule [7], MineSQL [8]. However none of the mentioned above solutions is adapted to a distributed data mining.

Proposed EwSQL query syntax allows simple and clear definition of data mining models features. Models can be built locally or in a distributed manner – using combining strategy phrase in a query (for a distributed data mining) is optional.

The basic form of proposed SQL enhancement for a distributed data mining is as follows:

```
<data_set> <model_type> <operation_type>
<model_name> [TARGET <target_attribute>]
[OBJECT <object_identifier>] WITH <model_params_list>
BY <analysis_space_attributes_list>
[COMBINE MODELS WITH <combining_str_params_list>]
```

Definitions of individual elements are as follows:

```
<data_set> ::= SELECT * || <fields_list> FROM <table>
<model_type> ::= PREDICT | CLUSTER | CLASSIFY | ASSOC
<operation_type> ::= TRAIN | TEST | APPLY
<model_params_list> ::= <algorithm> [, <algorithm_params>]
<combining_str_params_list> ::= <str> [, <str_params>]
```

Example EwSQL query:

```
SELECT * FROM persons CLASSIFY TRAIN model_p
TARGET class OBJECT ssn WITH (ID3) BY ALL_DIMS
COMBINE MODELS WITH S_1
```

In the above query we are building classification model named 'model_p' for the person set using ID3 algorithm (ID3 has no parameters). ALL_DIMS phrase denotes dimensions list consists of all person's attributes except 'ssn' and 'class', for example income, age, etc. For combining local models hypothetical strategy S_1 is used.

References

1. Ullman, J.D., Widom, J.: A First Course in Database Systems. Prentice-Hall, Inc., Englewood Cliffs (1997)
2. Hand, D., Mannila, H., Smyth, P.: Principles of Data Mining. The MIT Press, Cambridge (2001)
3. Gorawski, M., Pluciennik, E.: Analytical Models Combining Methodology with Classification Model Example. In: 1st IEEE International Conference on Information Technology, Gdansk, Poland (2008)
4. International Organization for Standardization (ISO). Information Technology, Database Language, SQL Multimedia and Application Packages, Part 6: Data Mining Draft Standard No. ISO/IEC 13249-6 (2003)
5. Han, J., Fu, Y., Wang, W., Koperski, K., Zaiane, O.: DMQL: A Data Mining Query Language for Relational Database. In: Proc. Of SIGMOD Workshop DMKD, Montreal, Canada (1996)
6. Imieliński, T., Virmani, A.: MSQL: A Query Language for Database Mining. Data Mining and Knowledge Discovery (1999)
7. Meo, R., Psaila, G., Ceri, S.: An Extension to SQL for Mining Association Rules. Data Mining and Knowledge Discovery (1998)
8. Morzy, T., Zakrzewicz, M.: SQL-like language for database mining. In: Proc. of the First East-European, Symposium on Advances in Databases and Information Systems - ADBIS, St. Petersburg (1997)

Construction and Querying of Relational Schema for Ontology Instances Data

Maciej Falkowski and Czeslaw Jedrzejek

Institute of Control and Information Engineering, Poznan University of Technology,
M.Sklodowskiej-Curie Sqr. 5, 60-965 Poznan, Poland
{maciej.falkowski,czeslaw.jedrzejek}@put.poznan.pl

Abstract. In this paper we present a method of storing semantic RDF instances data in relational databases. This method is based on the "table-per-class" approach, and present an algorithm of automatic relational schema building for ABox data. We also developed query processor that rewrites SPARQL queries to SQL queries and has some inferencing capabilities. Tests on our prototype system demonstrate that the rewritten queries can be answered by RDBMS in an efficient and scalable way.

1 Storing Semantic Data in Relational Databases

With the spreading of Semantic Web technologies an issue aroused to effectively store, process and retrieve large volumes of RDF-formatted data. Native RDF stores develop rapidly, but lack industry-level capabilities of existing relational databases, such as scalability, security, recovery and others. In this paper we describe a method of building and querying relational schemata designed to store ontology instances data. An impedance mismatch between relational data model and RDF data models can be overcome by simple three-column table, eg. Oracle's RDF support [2006]. More sophisticated approach is used in a RDF toolkit build on top of IBM's DB2[Z], where specialized schemas for TBox and ABox data can lead to increase efficiency while maintaining universality. Our approach goes further and forsakes the versatility for the efficiency. The main idea is to treat ontology as a data model and create specialized schemata for instances data described by that ontology. This can be done fully automatic. Generated schemata is based on the "table-per-class" approach, modified to take into account cardinality of properties. A property that is not constrained to be single-valued or has no domain gets a separate table, as relational model do not permit multi-valued columns. A property that is single-valued and has at least one domain is expressed by a column in domain(s) table(s). Having a class as a domain does not mean that objects of this class can have that property (like in OO languages). It means that every object having this property is of a domain class(es). Translating this behavior to relational schema, properties can be expressed as columns of domain classes tables. If single-valued property has more than one domain it will be stored in every domain table, forsaking storage inefficiency to retrieval efficiency. Cardinality constraints can be expressed

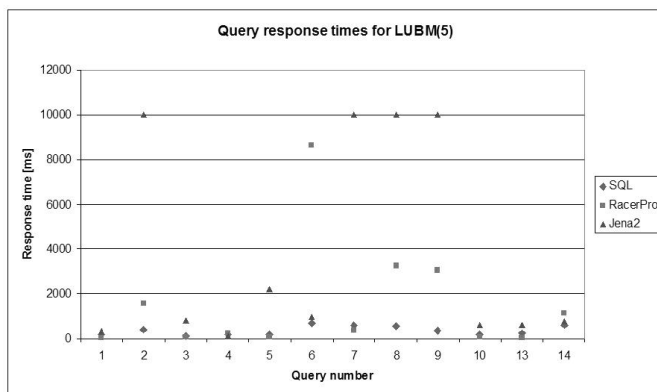


Fig. 1. LUBM(5) benchmark results

in two ways. Every property defined as functional (`owl:FunctionalProperty`) is constrained in every context to cardinality of 1. Besides that, class definition can contain property restrictions (for example, class `Orphan` can be defined as restricted to have zero `haveParent` properties). Resulting schema allows to store only instances data, not ontology itself and is vulnerable to ontology changes. Major changes lead to expensive schema redesign. Ontology is used by the query engine to formulate queries (i.e. to check if given property is stored as a table or as a column). We have also built a query processor for our schemata that relies on query rewriting and have some reasoning capabilities. Conducted tests with the LUBM benchmark [2005] proved that in some domains, when the main issue is fast data retrieval and ontology expressiveness is not crucial our approach performs very well. Efficiency depends highly on ontology characteristic, especially on single valued properties and overall number of classes. It is best suited for small to moderate ontologies and large volumes of data. Our future studies concentrate on using indices, views and materialized views to get even better efficiency and on methods of minimizing data redundancy.

References

- [2005] Guo, Y., Pan, Z., Heflin, J.: LUBM: A Benchmark for OWL Knowledge Base Systems *Web Semantics* 3(2), 158–182 (2005)
- [2006] Sharma, J.: An Introduction to the Oracle Database 10gR2 RDF storage and query model. In: *RDF, Ontologies and Meta-Data Workshop* (2006)
- [2008] Ma, L., Wang, C., Lu, J., Cao, F., Pan, Y., Yu, Y.: Effective and Efficient Semantic Web Data Management over DB2. In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1183–1194 (2008)

Evaluation of the Navigation through Image Parts in the ImageNotion Application

Andreas Walter¹, Gabor Nagypal², and Simone Braun¹

¹ FZI Research Center for Information Technologies, Information Process Engineering,
Haid-und-Neu-Straße 10-14, 76131 Karlsruhe, Germany

² disy Informationssysteme GmbH, Erbprinzenstr. 4-12, Eingang B, 76133 Karlsruhe, Germany
Andreas.Walter@fzi.de, nagypal@disy.net, Simone.Braun@fzi.de

Abstract. In our work on the ImageNotion methodology and tools, we apply semantic technologies on image archives. In this paper, we show evaluation results on our work on the user interface for semantic image search and expected navigation through image parts. We conducted an online survey with more than hundred participants. A unique feature of our evaluation is that our evaluators filled the survey based on a concrete, working semantic application, i.e., based on the publicly available online version of our system.

1 Evaluation Results

The ImageNotion methodology [12] is a visual methodology which supports collaborative, work-integrated ontology development, collaborative semantic annotation and visual semantic search. The ImageNotion application that implements the ImageNotion methodology is publicly available at www.imagenotion.com.

We conducted the evaluation by executing an online survey. We sent email invitations to mailing lists of currently running EU projects (such as Theseus, Mature-IP and IMAGINATION), to German and French image agencies, professional image searchers, historical archives, universities and companies. Altogether, we reached over 1.000 recipients with our email invitations. 137 users accepted the invitation and participated in our survey.

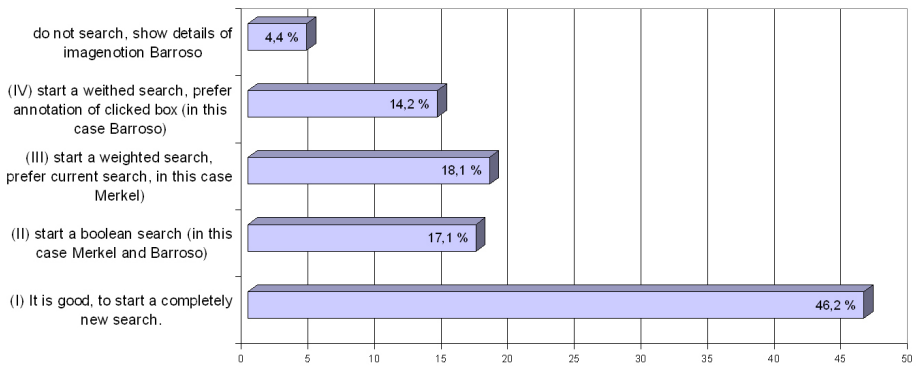
In the first part of our evaluation, we were interested in the general usability of semantic technologies for image search. The results are listed in Table 1.

The focus in this paper is what users expect when using the navigation through image parts. The task was to perform a semantic search for “Angela Merkel”. Based on the search result that displays thumbnail images of the current German chancellor, users were asked to open one of these images in a bigger window (displaying the so-called layout image). This image contains annotation boxes for the faces of “Angela Merkel” and “Manuel Barroso”

We asked the users what should happen when they click on the annotation box for “Manuel Barroso”. Fig. 1 shows, that more than the half of the users (53.8%) did not want to start a completely new search. Instead, 17.1% of the users preferred starting a boolean search. An even higher number, 32.3%, preferred a weighted search mechanism and only 4.4% of the users preferred not to search but to see the background information of the “Manuel Barroso” imagenotion.

Table 1. Yes/no questions of the evaluation

Question	Yes (%)	No (%)
Did you understand the notion of imagenotions?	95.6	4.4
Are refinement and cluster mechanisms useful for the navigation through image archives?	81.1	18.9
Do you have a problem with big number of annotation boxes on images?	53.4	46.6
Do you like the idea of filtering annotation boxes using topical groups?	87.9	12.1

**Fig. 1.** Expected search when clicking on annotation boxes

We found out that users expect more from the navigation through image parts feature than simply starting a new search request, which is the solution of current state of the art systems, including the current state of ImageNotion. We plan to implement the various query refinement possibilities that users requested in future versions of our system.

References

1. Walter, A., Nagypál, G.: Imagenotion - methodology, tool support and evaluation. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 1007–1024. Springer, Heidelberg (2007)
2. Walter, A., Nagypál, G.: Efficient integration of semantic technologies for professional image annotation and search. In: Proc. of the IADIS International Concerence e-Society, Portugal, April 8-12, 2008, IADIS-Press (2008)

Instance-Based Ontology Matching Using Regular Expressions

Katrin Zaiß, Tim Schlüter, and Stefan Conrad

Institute of Computer Science
Heinrich-Heine-Universität Düsseldorf
D-40225 Düsseldorf, Germany

{zaiss, schlueter, conrad}@cs.uni-duesseldorf.de

Many algorithms dealing with the matching task have been proposed in the past, most of them not considering instances. Only a few existing systems like [EM07] or [DMDH04] use the information provided by instances in their matching algorithms. Due to the fact that ontologies offer the possibility to model instances within the ontology, these should definitely be used to increase the accuracy of the matching results. Additionally, the set of instances probably provides more information about the meaning of a concept than its label. Thus, we propose a new instance-based ontology matcher which should extend existing libraries to enhance the matching quality.

The basic idea behind our approach is to consider the content of an ontology and not mere the outer form (e.g. the name of a concept or the data type of an attribute). Our algorithm scans a sample of the instances in order to describe the contents of every concept through a set of regular expressions. These sets can easily be converted into single vectors, each of them representing a concept. These concept vectors establish a basis for calculating a similarity value (e.g. the cosine similarity) to discover similar concepts and thus to match ontologies. The flow of our matching process is shown in figure 1.

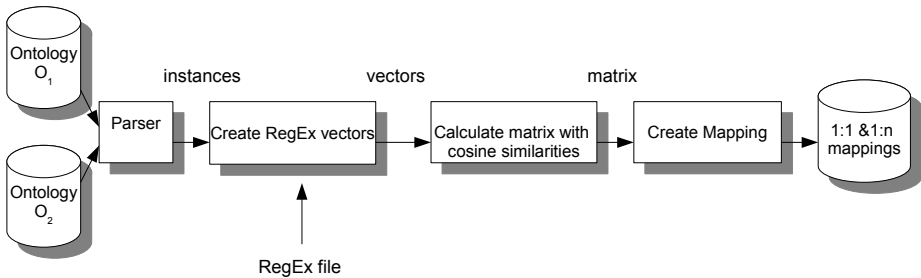


Fig. 1. Matching Process

Building the RegEx Vectors

A domain expert has to create a list of regular expressions that fits to (the instances of) the ontologies for a special domain. The instances are compared to this RegEx list, and the first fitting regular expression is assigned to the instance, whereas the regular expression that is assigned to the majority of the instances belonging to a certain attribute is assigned to this attribute.

RegEx list

```
.*(((c|C)onference)|((w|W)orkshop)|((s|S)ymposium)|((t|T)utorial)).* // conferenz etc.
(19(\d)(\d)|(200(\d)) // year (1910-2009)
((([A-ZÖÜÄ][a-zäöüß]{0,4}(\.))?(s)?)|([A-ZÖÜÄ][a-zäöüß]*(s)?))+ // name (shortname, etc.)
[\w-\.]++@([\w-]+\.)+[\w-]{2,4} // email
[\d\s]+ // simple number
[\wüöäåöß\-\s]+ // text with some additional character
...
```

Concept "Unpublished"

ID	AUTHOR	EMAIL	TITEL
168	I. N. Bronnshtein	master@math.org	YAH - Yet Another Handbook
255	Mickey Mouse	mickey@disney.com	Cinderella 2 - The Beginning
...

$$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ \dots \end{pmatrix}$$

Attribute	Regular Expression
ID	(\d)++
AUTHOR	(((A-ZÖÜÄ[a-zäöüß]{0,4}(\.))?(s)?) ([A-ZÖÜÄ][a-zäöüß]*(s)?))+
EMAIL	[\w-\.]++@([\w-]+\.)+[\w-]{2,4}
TITLE	[\wüöäåöß\-\s]++

Fig. 2. Creating the RegEx vector

In order to avoid a false assignment in this process, the regular expressions have to be ordered from very specialized at the beginning of the list to more universal at the end.

The RegEx vector for a concept c is a vector $v \in \mathbb{N}^d$ with d being the number of different regular expressions. In every component v_i it contains the number of assignments of the i -th regular expression in RegEx list l to an attribute in c . An illustration of the whole process is given in figure 2.

Creation of Candidate Mapping

For determining a mapping we need the cosine similarities of all possible concepts pairs and a certain predefined threshold. 1 : 1 mappings are created by gradually finding the concepts pairs with the highest similarity assuming that every concept has at most one best-fitting matching partner. The whole process is repeated until there are no more similarity values above the threshold, i.e. there are probably no more 1 : 1 mappings.

To find 1 : n mappings we regard the remaining concepts and compute the cosine similarity between one concept vector and the sum of n other vectors. A resulting similarity value above the threshold is a hint for a 1 : n mapping.

References

- [DMDH04] Doan, A., Madhavan, J., Domingos, P., Halevy, A.Y.: Ontology Matching: A Machine Learning Approach. In: Handbook on Ontologies, pp. 385–404. Springer, Heidelberg (2004)
- [EM07] Engmann, D., Maßmann, S.: Instance Matching with COMA++. In: Datenbanksysteme in Business, Technologie und Web (BTW 2007), Workshop Proceedings, Aachen, Germany, März 5-6 (2007)

ADI 2008 PC Co-chairs' Message

Welcome to the First International Workshop on Ambient Data Integration (ADI 2008). The workshop was held in conjunction with the On The Move Federated Conferences and Workshops (OTM 2008), November 9-14, 2008 in Monterrey, Mexico.

This workshop is concerned with the subjects that are relevant for the success of data integration systems, such as distribution and conceptualization. The five accepted papers propose an interesting mix of conceptual, technical and application-oriented solutions.

On the subject of conceptualization, the paper by Lilia Munoz et al. provides a new approach for modeling ETL processes that are used to integrate heterogeneous data into a repository. Bernhard Volz's paper considers typical data integration steps in the context of process management. Technical aspects of schema matching are studied in the paper by Amar Zerdazi et al., where the authors introduce a solution based on the notion of structural node context. Finally, two papers are related to applicational aspects. The paper by Michael Hauhs presents data integration requirements in ecosystem research; the paper written by Silvia Bindelli et al. presents the TagOnto system, which is designed to combine social tagging and an ontology-based approach for semantic retrieval.

We would like to thank Avigdor Gal for his keynote lecture on data integration, the authors for their submissions, the program committee members for their excellent work, and the conference organizers for their great support in setting up the workshop.

November 2008

Christoph Bussler
Olivier Curé
Stefan Jablonski

Modelling ETL Processes of Data Warehouses with UML Activity Diagrams*

Lilia Muñoz¹, Jose-Norberto Mazón², Jesús Pardillo², and Juan Trujillo²

¹ Lucentia Research Group, Dep. of Information Systems, Control, Evaluation and Computing Resources, University Technological of Panama, Panama

`lilia.munoz@utp.ac.pa`

² Lucentia Research Group, Dep. of Software and Computing Systems, University of Alicante, Spain

`{jnamazon,jesuspv,jtrujillo}@dlsi.ua.es`

Abstract. *Extraction-transformation-loading* (ETL) processes play an important role in a *data warehouse* (DW) architecture because they are responsible of integrating data from heterogeneous data sources into the DW repository. Importantly, most of the budget of a DW project is spent on designing these processes since they are not taken into account in the early phases of the project but once the repository is deployed. In order to overcome this situation, we propose using the *unified modelling language* (UML) to conceptually model the sequence of activities involved in ETL processes from the beginning of the project by using *activity diagrams* (ADs). Our approach provides designers with easy-to-use modelling elements to capture the dynamic aspects of ETL processes.

Keywords: ETL, UML, activity diagrams, modelling, processes.

1 Introduction

In the nineties, Inmon [1] coined the term *data warehouse* (DW) as a “collection of integrated, subject-oriented databases designated to support the decision support function”. Specifically, a DW is *integrated*, because data are collected from heterogeneous sources (legacy systems, relational databases, COBOL files, etc.) to adapt them for decision making. Importantly, the integration of these sources is achieved in the DW domain by defining a set of *extraction-transformation-loading* (ETL) processes. These processes are responsible for extracting data from heterogeneous sources, transforming their data into an adequate format (by conversion, cleaning, etc.) and loading the processed data into the DW.

Designing ETL processes is extremely complex, costly and time consuming [2]. However, it has been broadly argued in the literature that ETL processes are one of the most important parts of the development of a DW [10].

* Supported by Spanish projects ESPIA (TIN2007-67078) and QUASIMODO (PAC08-0157-0668). Lilia Muñoz is funded by SENACYT and IFARHU of the Republic of Panama. Jose-Norberto Mazón and Jesús Pardillo are funded under Spanish FPU grants AP2005-1360 and AP2006-00332, respectively.

Shilakes [7] reports that ETL and data cleaning tools are estimated to cost at least one third of effort and expenses in the budget of a DW, while [8] mentions that this number can rise up to 80% of the development time in a DW project. Currently, specialised tools provided by DBMS vendors such as [3,5,4], are widely used for specifying ETL processes in a DW environment. Unfortunately, these tools have some drawbacks since they lack in a conceptual modelling perspective: (i) lack of specificity and expressiveness, (ii) dependency on the target platform, (iii) highly complex configuration and setup. Furthermore, the high price of acquisition and maintenance of these tools makes that many organisations prefer to develop their own ETL processes by means of specific programs. Therefore, this scenario can make the design of ETL processes difficult for the integration of heterogeneous data sources in the DW.

To overcome these drawbacks, in recent years, several proposals have been defined for the conceptual modelling of ETL processes [10,11,13,12,14]. They advocate them from the perspective of the sources, and their transformation processes of those sources. However, they only propose static structures to model ETL processes, which do not allow to evaluate the behaviour of the designed ETL processes. Furthermore, they do not define formal mechanisms for representing special conditions, *e.g.*, sequence of the control flows or temporal restrictions. Finally, some of these proposals are not formally integrated in a concrete framework, thus providing only partial solutions and making difficult their application when a disparate set of sources is being integrated. Keeping in mind these considerations and the need of new modelling elements to represent special conditions of ETL processes, this paper proposes the developing of a *conceptual modelling framework*, that allows us to clarify the behaviour of an ETL process. To this aim, we take advantage of the high expressivity of the *unified modelling language* (UML) [9], specifically, the *activity diagrams* (ADs). The UML ADs are behavioural diagrams used to capture the dynamic aspects of a system. In this sense, we designed a set of *modelling elements* of AD to represent the activities involved in ETL processes, those will allow us to model the behaviour of a process and the seamless integration of the design of ETL processes together with the DW conceptual schema. This paper is organised in the following way. In Section 2, we present the related work. In Section 3, our proposal based on UML ADs for the conceptual modelling of ETL processes of DW is presented. In Section 4, we state a concrete example of application of our proposal. Finally, in Section 5, we encompass the main conclusions and future work.

2 Related Work

Conceptual Models of ETL Processes. Conceptual modelling of ETL processes have been developed from several perspectives: generic modelling [10], development methodologies [11] and ontology-based [14]. Although these approaches are interesting (becoming a milestone in ETL design), they lack in providing mechanisms for conceptually representing some important issues, such as temporal conditions or behaviour, and dynamic aspects of ETL processes. Furthermore,

these approaches are not integrated in a development framework nor use standards for their development. Just in [13], Luján *et al.* suggest an extension of UML within a new modelling framework, namely *data mapping*, which grants to represent transformation rules among the necessary attributes to model the ETL processes at the conceptual level. With this proposal, they try to solve the data processing problem in a very low granular level. In turn, Trujillo & Luján [12] suggest another modelling framework based on UML class diagrams which allow designers to decompose a complex ETL process into a set of simple processes. In spite of their benefits, these proposals model static structures that do not permit to evaluate the dynamic aspects, neither the behaviour of an ETL process.

Logical and Physical Modelling of ETL Processes. Aside from the commercial tools [3,5,4], great research efforts [15,16,18,17,20] has been done to consider the development of tools which are capable of modelling and executing ETL processes. Nevertheless, these proposals are not integrated into a global DW framework, thus appearing problems of interoperability and integrity¹ [6].

Moreover, we base our proposal on [12], in which the authors advocate for the use of UML class diagrams, allowing to decompose ETL processes in smaller logical units. Each activity is presented through a stereotyped class, enabling developers to easily design ETL processes through different detail levels. We consider that this proposal offers a very clear and high-level overview of ETL processes. Nevertheless, this proposal lacks in mechanisms to represent time constraints, dynamic aspects, and sequencing of control flows. In order to solve these problems, we use ADs. To this aim, the modelling of activities places emphasis on the sequence and conditions for the coordination of the behaviour and its classification [9]. Besides, the activities are focused on representing sequences, conditions, inputs and outputs to invoke other behaviours [22].

3 UML Activity Diagrams for Modelling ETL Processes

This section describes how to conceptually model ETL processes for DWs by focusing on UML ADs. Our proposal consists of a *modelling framework* for ETL processes, built through a set of reusable and parameterised modelling elements. They are developed using the metaclasses, *i.e.*, primitive UML modelling elements, of the ADs. In turn, these elements allow to represent dynamic aspects and behaviour, to arrange the control flow and incorporate restrictions of temporality (*e.g.*, time that a process takes to be executed). The frame of reference for the development of our approach is the proposal described by Trujillo & Luján [12], in which the modelling through UML class diagrams is presented. Nevertheless, these class diagrams present a high level view, so the designer has a very clear perspective from the beginning. In a second step of refinement, we will use the provided modelling elements to have a lower level vision of dynamic aspects and behaviour of the modelled ETL processes.

¹ We refer the interested reader to [19], for a detailed study about the efforts of investigation in the scope of ETL tools.

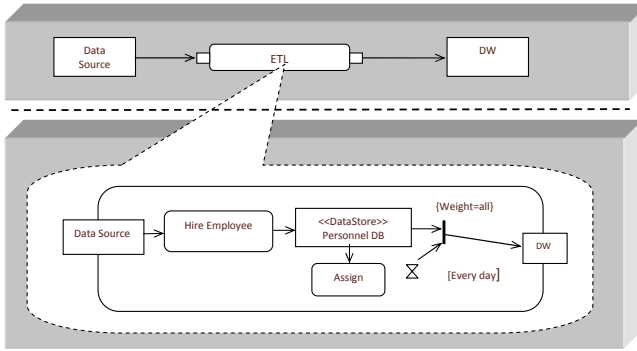


Fig. 1. The two levels of the proposed modelling framework

In Fig. 1, we describe the proposed modelling framework. At the highest level of abstraction the elements of a ETL process are specified. On the left-hand side, we represent the data sources that are involved in an ETL process (*e.g.*, relational databases, files, etc.) which is represented in the middle of the figure. The DW is specified on the right-hand side and comprehends the target sources (*e.g.*, fact tables or dimensions tables). Furthermore, at the lower level of abstraction, we present the flow of sequence of an ETL process. This flow uses metaclasses of UML ADs which provide us with high expressivity for modelling. On its left-hand side, data sources are specified. Subsequently, the process flow is described and, on the right-hand side, the DW comprehends the target sources.

In turn, for our framework development, we draw on the whole activities presented in [12], since they are representative operators for the conceptual modelling of ETL processes. Next, we present them together with a brief description:

Aggregation: It aggregates data based on some criteria.

Conversion: It changes data type and format or derives new data.

Filter: It filters and verifies data.

Incorrect: It redirects incorrect data.

Join: It joins two data sources related to each other with some attributes.

Loader: It loads data into the target of an ETL process.

Log: It logs activity of an ETL mechanism.

Merge: It integrates two or more data sources with compatible attributes.

Surrogate: It generates unique surrogate keys.

Wrapper: It transforms a native data source into a record based data source.

The elements of our proposal are designed by using a customisable and instantiable template (see Fig. 2), that includes a generic model of a reusable activity in an ETL process. It is defined on the basis of a set of elements of type *ParameterableElement* [9], which are part of the modelling element. These elements are abstractions that represent the various modelling elements that make up a concrete instance. The parameters of the construct ($P1$, $P2$, $P3$), are used

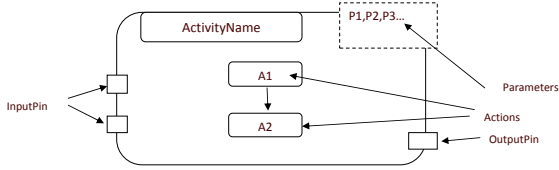


Fig. 2. Template employed for specifying ETL processes at the conceptual level

to specify certain indicators *ParameterableElement* which may vary from one to another instance. They can vary depending on the required features and attributes. Moreover, we represent the *Actions*, *InputPin*, and *OutputPin* [9] that are part of the diagram of activities. Due to the space constraints, in the following sections, we present the modelling elements for the most significant tasks which allow designer to specify the most complex aspects of ETL process.

3.1 Aggregation Activity

The *Aggregation* activity adds data based on some criteria (e.g., *SUM*, *AVG*, *MIN*, *MAX*, *COUNT*). This activity is useful to increase the level of aggregation of the data sources. Therefore, the partial aggregation under certain criteria is a technique commonly used in DWs in order to facilitate the complex analysis. At the same time, it reduces the DW size and increase the query performance [2].

In Fig. 3, we present the modelling elements for the *Aggregation* activity template (a) and its instantiation (b). For example, on the right-hand side, the template is instantiated to model the activity. Initially, we have the sales of tickets online, but the total of daily sales in the DW is needed. The work flow is: the attributes are extracted (*IdProducts*, *Name*, *Price*, *Quantity*, etc.), from the data source (*Sales*), and then, the key attributes are verified (action *Verify*). For doing that, the operation is carried out to calculate the total of sales ($Total = (SUM (Quantity * Price))$), as input parameter we use the criteria *SUM* (action *Operation*). Subsequently, the elements are grouped by *IdProduct* & *Date*. For this aim, we use the operation *GroupBy* (action *Function*). The result of this action is temporarily stored in a table (*Table Temp*) by means of the metaclass *DataStoreNode* [9]. Finally, the information is sent to the DW.

3.2 Conversion and Log Activities

The *Conversion* activity is used for changing formats and type of data or creating new ones from existent data. The syntax for this activity is the following: `target = attribute Function (source attributes)`, where *Function* is some type of function that can be applied to the attributes of the data source. In our proposal, we model the *Log* activity, like an activity that sends signs to a *DataStoreNode* metaclass when an activity has been carried out.

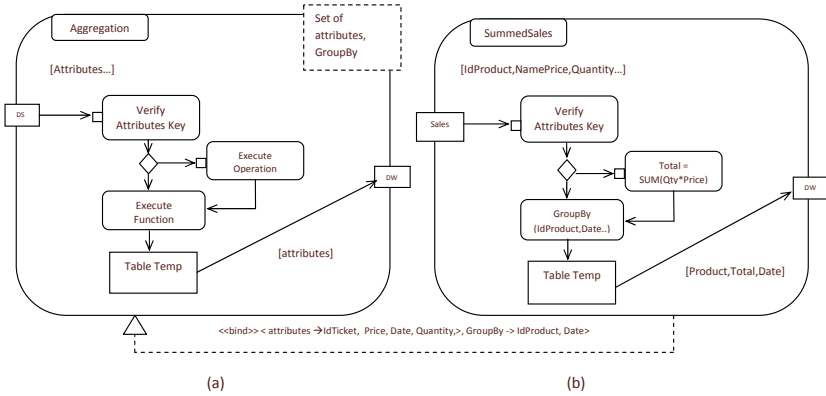


Fig. 3. Aggregation activity template (a) and example instantiation (b)

In Fig. 4(a), it is presented the template for the *Conversion* and *Log* activities. For their instantiation (b), we present a conversion example: concatenation of *Name* and *Surname*. The activity flow is started by extracting the attributes (*IdCustomer*, *Name*, *Address*) of the data source (*Customers*). Once verified the attributes to be converted (action *Verify*), they are transformed according to the conversion criteria, for this we use the metaclass *ObjectNode* [9] which is specified in a stereotyped note as <<Transformation>> (*Name = Concatenate (Name, “ ”, Surname)*). Finally, the converted objects pass for a flow of control *ForkNode*, which allows to send a token as a sign that the operation of conversion has been carried out. For this, we use the metaclass *SendSignalAction*, a call *Notify* and another token indicating the data sending to the DW.

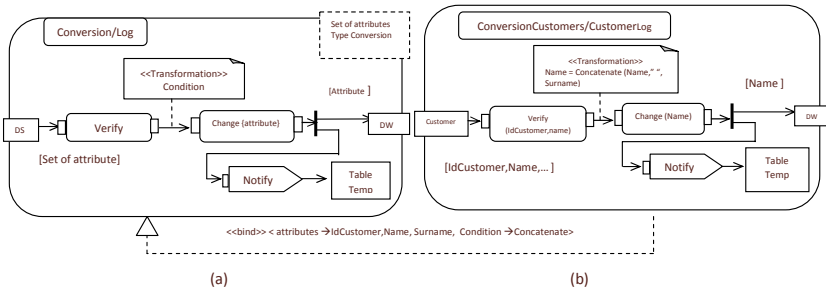


Fig. 4. Conversion and Log activities template (a) and example instantiation (b)

3.3 Filter Activity

The *Filter* activity filters the unwanted data and verifies the accuracy of the data based on some restrictions. In this way, the activity allows the designer to

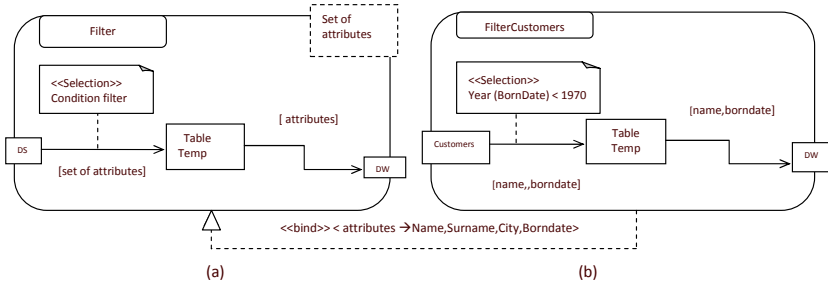


Fig. 5. *Filter* activity template (a) and example instantiation (b)

charge only the required data or those which fulfil a given quality level in the DW. Some of the tasks of this activity are: to control the null values, the missing values, or the values out of range. The data that do not fulfil the conditions can be sent to the incorrect processes activity *Incorrect*.

The template of the *Filter* activity is presented in Fig. 5 (a). For instance, regarding to its instantiation (b), we describe an example where clients born before 1970 are filtered. For this, the attributes (*Name*, *Surname*, *BornDate*, *City*) are extracted of the data source (*Customers*). Subsequently, the object node selects the input attributes according to the criteria specified in a stereotyped note as «*Selection*». In this case the object node selects only those clients born before 1970. The result of this activity can be stored temporarily in a table *Table Temp* which is modelled by the metaclass *DataStoreNode*.

3.4 Join Activity

The *Join* activity is used to join a disparate set of data sources through attributes (which are defined through constraints). The function for the *Join* would be $f(x) = \text{Join}(\text{ConditionalExpression})$, where *ConditionalExpression* defines the union of the data sources. On the other hand, the activity *Join* can be of a particular type (*InnerJoin*, *LeftJoin*, *RightJoin*, *FullJoin*). For example, *FullJoin* includes all the records of the two data sources.

In Fig. 6, we present the template for the *Join* activity (a) and an example instantiation (b). This activity may have two or more data sources as input. In our example, we shall use only two sources (*Cities*, *States*). In turn, from the source *Cities* we extract the attributes (*IdCity*, *Name* and *State*) and from the source *States* (*IdState*, *Name*). It is required to do a *Join* of *State* and *IdState*. For this, the attributes are verified (action *Verify*). For doing that, the operation *Join LeftJoin (State = IdState)* is carried out. We use the metaclass *JoinNode* that permits to synchronise multiples concurrent flows. Finally, the *Join* is confirmed; for this, we use the metaclass *SendSignalAction* which we call (*Notify*), and the information is stored in a table *Table Temp*.

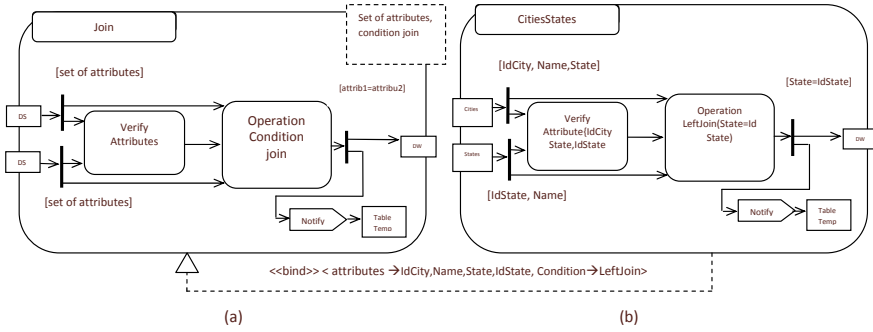


Fig. 6. Join activity template (a) and example instantiation (b)

4 Example Application for Designing ETL Processes

In Fig. 7 we represent a loading process for the DW. It has the online sales of tickets (data source), however, the total of daily sales in the DW is needed (the fact table *Sales* is on the right-hand side of Fig. 7). For this, we have a data source (*Sales*) that contains the following attributes (*IdTicket, IdProducts, Name, Description, Price, Quantity, Discount, Date*). It is needed to group the sales by *IdProduct* and *Date*, we use the activity *Summedsales*. The flow of this

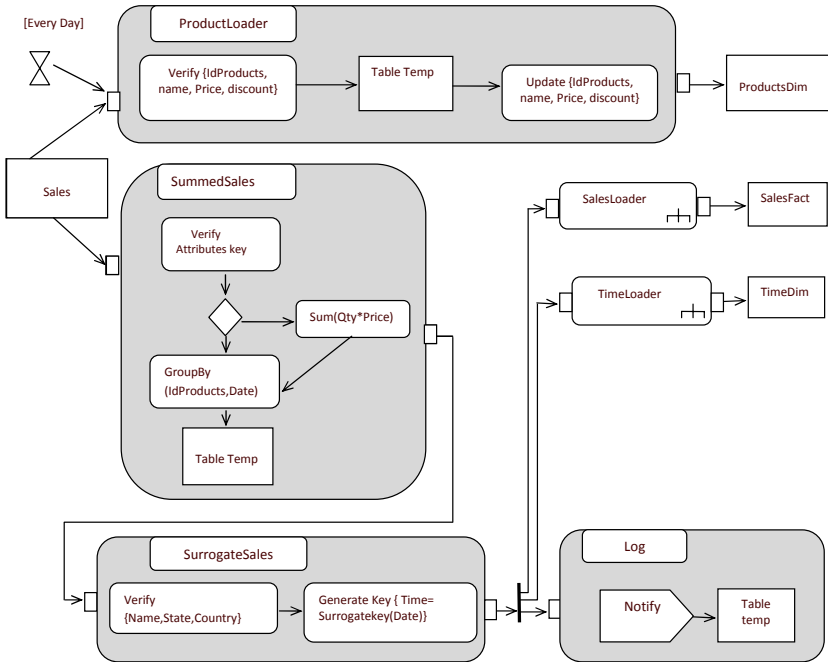


Fig. 7. Example ETL process modelled by the proposed framework based on ADs

activity is modelled in this way: once the key attributes are verified in the action *Verify*, and, because the total of daily sales is requested, the *Total* operation, $Total = SUM(Quantity * Price)$, is performed. Subsequently, the function *GroupBy* is applied. It will allow us to have the sales grouped by *IdProducts* and *Date*. This information is stored in a table *Table Temp*. The information of the table *Table Temp* is used for the activity *SurrogateSales* to generate an alternate key (*Time* based on an attribute *Date*). This key is used to establish a relation between the fact table *Sales* and the dimension *Time* (*Sales* contains a foreign key *Time*). A *Log* activity notifies if the key has been generated. Besides, the summarised sales are loaded in the fact table *Sales*, this operation is carried out with the activity *SalesLoader*, for the sake of understanding, this activity is not exploded in the example and it is represented with a metaclass *CallBehaviorAction*. On the other hand, it is needed to load the list of products into the dimension *Products*. For this, we use the activity *ProductsLoader* that verifies the attributes are extracted of the table *Sales*, and it stores them temporarily in a table *Table Temp* in which they are updated and subsequently loaded in the dimension *Products*. A temporal condition has been added to this activity, which indicates that the load should be carried out daily. The activity *TimeLoader* has not been exploded and it is represented by a metaclass *CallBehaviorAction*.

5 Conclusions and Future Work

The improvement presented in UML 2.0 concerning ADs allows designers to use this kind of diagram to represent processes and define new mechanisms, such as time constraints, sending signs, or sequencing flow control. The previous versions of the UML standard modelled these elements with notes, thus overlooking their formal treatment in a software development process.

Taking the benefits described in the preceding paragraph, we have presented a set of modelling elements based on ADs to model the behaviour of ETL processes. At the same time, these elements allow to represent temporality constraints (*e.g.*, time that a process takes to be executed) and arrange the control flow of a given process. In this context, the main benefits of our work are: (i) the visual modelling of complex processes, also enabling the interoperability with other diagrams, (ii) providing a set of templates that facilitates the description and reusability of common concepts in ETL processes, and (iii) facilitating the integration of their design in a global and integrated approach for DW development.

Our immediate future works include, for instance, studying the possibility of representing through use cases the modelling ETL processes, and applying some model transformation languages to generate code from the presented framework, and thus, integrating our proposal into a DW development framework, *e.g.* [23].

References

1. Inmon, W.: Building the Data Warehouse. Wiley, Chichester (1992)
2. Kimball, R., Caserta, J.: The Data Warehouse ETL Toolkit. Wiley, Chichester (2004)

3. Oracle: Oracle Warehouse Builder 10g, <http://www.oracle.com>
4. Microsoft: SQL Server 2005 Integration Services (SSIS), <http://technet.microsoft.com/enus/sqlserver/bb331782.aspx>
5. IBM: WebSphere DataStage, <http://www.ibm.com>
6. Jarke, M., Lenzerini, M., Vassiliou, Y., Vassiliadis, P.: *Fundamentals of Data Warehouses*. Springer, Heidelberg (2000)
7. Shilakes, C., Tylman, J.: *Enterprise Information Portals*. Enterprise Software Team, <http://sagemaker.com/company/downloads/eip/indepth.pdf>
8. Demarest, M.: The politics of data warehousing, <http://www.hevanet.com/demarest/marc/dwpol.html>
9. OMG: Unified Modelling Language. Version 2.0 (2005), <http://www.omg.org>
10. Vassiliadis, P., Simitsis, A., Skiadopoulos, S.: Conceptual modeling for ETL processes. In: DOLAP (2002)
11. Simitsis, A., Vassiliadis, P.: A Methodology for the Conceptual Modeling of ETL Processes. In: CAiSE Workshops (2003)
12. Trujillo, J., Luján, S.: A UML Based Approach for Modeling ETL Processes in Data Warehouses. In: ER, pp. 307–320 (2003)
13. Luján, S., Vassiliadis, P., Trujillo, J.: Data Mapping Diagrams for Data Warehouse Design with UML. In: Atzeni, P., Chu, W., Lu, H., Zhou, S., Ling, T.-W. (eds.) ER 2004. LNCS, vol. 3288, pp. 191–204. Springer, Heidelberg (2004)
14. Skoutas, D., Simitsis, A.: Designing ETL processes using semantic web technologies. In: DOLAP, pp. 67–74 (2006)
15. Vassiliadis, P., Simitsis, A., Skiadopoulos, S.: On the Logical Modeling of ETL Processes. In: CAiSE, pp. 782–786 (2002)
16. Vassiliadis, P., Simitsis, A., Skiadopoulos, S.: Modeling ETL activities as graphs. In: DMDW, pp. 52–61 (2002)
17. Vassiliadis, P., Vagena, Z., Skiadopoulos, S., Karayannidis, N., Sellis, T.: ARKTOS: towards the modeling, design, control and execution of ETL processes. *Information Systems*, 24 (2001)
18. Simitsis, A., Vassiliadis, P., Terrovitis, M., Skiadopoulos, S.: Graph-Based Modeling of ETL Activities with Multi-level Transformations and Updates. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2005. LNCS, vol. 3589, pp. 43–52. Springer, Heidelberg (2005)
19. Simitsis, A., Vassiliadis, P., Skiadopoulos, S., Sellis, T.: *Data Warehouse Refreshment*. In: *Data Warehouses and OLAP: Concepts, Architectures and Solutions*, IRM Press (2006)
20. Tziouvara, V., Vassiliadis, P., Simitsis, A.: Deciding the physical implementation of ETL workflows. In: DOLAP, pp. 49–56 (2007)
21. Simitsis, A., Vassiliadis, P., Sellis, T.: State-Space Optimization of ETL Workflows. *IEEE Trans. Knowl. Data Eng.* 17(10), 1404–1419 (2005)
22. Bock, C.: UML 2 Activity and Action Models, Part 2: Actions. *Journal of Object Technology* 2(5), 41–56 (2003)
23. Mazón, J.-N., Trujillo, J., Serrano, M., Piattini, M.: Applying MDA to the development of data warehouses. In: DOLAP, pp. 57–66 (2005)

Implementing Conceptual Data Integration in Process Modeling Methodologies for Scientific Applications

Bernhard Volz

University of Bayreuth, Bayreuth, Germany
bernhard.volz@uni-bayreuth.de

Abstract. Process management is a widely applied methodology for describing and solving complex application scenarios. Also process management is emerging in scientific domains (Scientific Workflows). Scientific domains are characterized by demanding data integration tasks where data have to be integrated stemming from heterogeneous sources. With respect to the importance of data integration many process modeling methodologies do not provide appropriate conceptual paradigms for specifying and enacting these kinds of tasks. Instead data integration is often treated as a normal work step within the flow of actions. However, this makes it difficult to cope with this data integration in a clean conceptual way. In this paper we will thus demonstrate how data integration tasks can be integrated in a process modeling language on a conceptual level using the Perspective Oriented Process Modeling approach and DaltOn, a framework for data transportation and semantic integration based on ontologies. We will further argue that data integration does not only affect data but instead is a more cross-cutting concern that has a certain impact on the definition of other building blocks of processes such as the operational and organizational perspective.

1 Introduction

The integration of data is an imminent need in complex (scientific) applications such as the climate prediction or the analysis of an unknown protein sequence where data stemming from diverse heterogeneous sources must be incorporated. Another major interest in scientific applications is a well structured description of the overall procedure – for which process modeling in general is a well-accepted means and many scientific workflow systems appeared in the past years (e.g. Kepler [10], Taverna [4] etc.). However process modeling languages do not contain special constructs for data integration nor do they envisage means for describing this kind of tasks. Instead, data integration is usually modeled as a normal step within a process; as any other step, these tasks use services for performing their job, usually hand-written wrapper applications.

Of course one can argue that this works well, because it is currently a wide spread solution. But both, the missing integration into the process modeling languages and the application of single wrappers, come with some pitfalls which become more and more eminent. Basically these issues are tightly bound to the rising number of transformations necessary within a process. If the number of (data transformation) steps grows, process models become more and more complex and

are harder to comprehend since the data integration steps conceal the “normal” (analytical) process. Second the number of wrapper components needed to execute all these transformations also rises since usually wrappers convert data from one system to another but are not generalized and thus hard to adapt to a new integration case.

In contrast to the application of wrapper applications and the modeling of integration tasks as normal steps within a workflow we propose an integration of these tasks within a process modeling methodology. Process modeling is used for describing complex applications as above. They can contain data integration tasks but information necessary for carrying them out is hidden from the user. The actual integration tasks are then enacted by a special framework which is based on ontological descriptions of data sources and sinks. This way, the complexity of the overall process is not increased by the integration tasks. Secondly, the hand-written and hard to maintain wrapper applications are replaced with a generalized framework. Another reason why a framework based on ontologies is tempting is data provenance; in many scenarios the ability to trace an instance of an application is a must. Then it is sufficient for the implementation of data provenance to store the original data, the involved ontologies and the actions performed on the data to be able to reproduce the execution of the original instance – even the application is no longer available.

We chose Perspective Oriented Process Modeling (POPM) [5] as process modeling method since it separates different aspects of a process into so-called perspectives; second the foundation of POPM is a flexible meta model that enables users to enrich the modeling language easily. The ontology based data integration framework is called DaltOn (“Data Logistics with Ontologies”) [3, 7]. Due to the general character of POPM we also consider the presented approaches to be applicable also in other fields beside scientific workflows (e.g. in business processes). The remainder is structured as follows: Section 2 explains aspects of conceptual data integration in process modeling. POPM and DaltOn are explained in more detail in Section 3; Section 4 shows how the structure of the second section can be implemented using POPM and DaltOn. Section 0 summarizes related work and Section 0 concludes this publication.

2 Aspects of Conceptual Data Integration in Process Modeling

The following section describes the aspects of data integration on a conceptual level. It also demonstrates why ontology based data integration is an important means for performing data integration.

Fig. 1 shows, how data integration can be structured (the hierarchy follows a classification for model transformations presented in [11]). In general, data integration contains two areas of interest, here called “spaces” – a Technological Space and a Domain Space. The Technological Space concerns the syntax of data (e.g. the format) whereas the Domain Space defines the semantics of data. In order to perform data integration these two spaces need to be taken into account – data formats might have to be adjusted as well as the semantics (e.g. unit conversions are needed).

Transformations in the technological space are again subdivided into two groups – the Endogenous and Exogenous Transformations. An Endogenous Transformation is a transformation which does not cross two spaces whereas each space is characterized

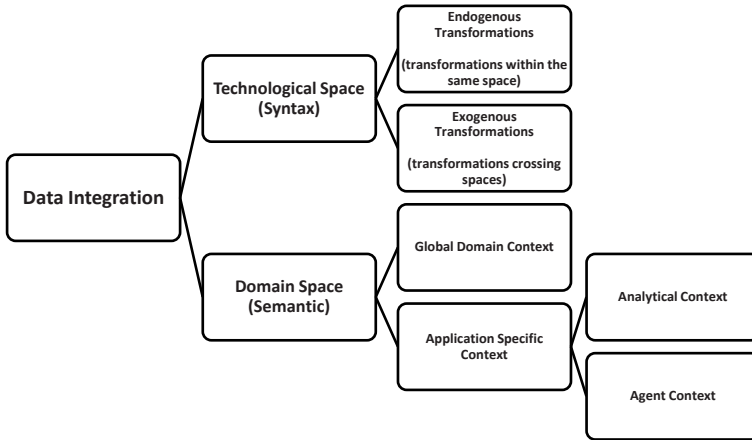


Fig. 1. Syntactic and semantic aspects of data integration

by its meta model. Thus converting a Microsoft Word document into an OpenOffice document is an endogenous transformation since both formats are based on XML and the technological space “XML” (characterized by a meta model that describes XML but not the single formats) is not left. In contrast a conversion from DOCX (Word 2007, XML) into DOC (Word 2003, binary format) is an exogenous transformation since space boundaries are crossed. [11] goes even further in refining the exogenous transformations in horizontal (same level of abstraction) and vertical transformations (level of abstraction is changed).

In contrast to the technological space which deals with syntactical data integration, the domain space concerns only the semantics. Such a space is not characterized by a meta model but by the domain of an application (e.g. biology, ecology etc.). It contains a complete description of the knowledge of this domain in the Global Domain Context – at least in theory. In practice, it is a highly complicated task to describe the knowledge of a domain using an ontology since the interests of stakeholders often differ significantly. However, different applications in a domain can be isolated such that it becomes possible to describe the (data) semantics of one application in the so-called Application Specific Context (ASC). Since an application concerns usually many users (“agents”) and tools, the ASC can be further subdivided into the Analytical Context and the Agent Context. The Analytical Context describes how a specific tool perceives data, as does the Agent Context for agents. Of course the ontologies of the Analytical and Agent Contexts must be somehow compatible with each other as both are placed in the same ASC. A linking element could be the ontology specified in the ASC – since it describes a common understanding for the whole application, all other subordinate ontologies must be compatible to it as well. If the ontologies are compatible to each other, semantic transformations can be performed –ontologies which are not related cannot be used for transforming data.

The benefit of following this approach is a better structure of the whole integration task; information about the syntax of data (i.e. descriptions of data formats) is separated from information about the perception of data by a domain, an application and

an agent. Furthermore, by introducing the Analytical and Agent Contexts differences in the understanding of data in between agents and analysis applications can be captured (for instance an agent can apply the unit “mph” whereas an analysis application is always using “km/h”). If both spaces are present, data integration can be performed on a conceptual basis – what is missing is the information where data resides and which data is to be used (i.e. criteria for extraction, insertion and filtering). But this information is, from a conceptual point of view, primarily not interesting. Thus we will postpone explanations about this information until Section 4.

3 Perspective Oriented Process Modeling and the DaltOn Integration Framework at a Glance

This section gives an introduction into Perspective Oriented Process Modeling methodology (POPM) and the DaltOn framework for ontology based data integration. We will be using both methodologies in order to provide an outline of the implementation of the above mentioned approach in Section 0.

Perspective Oriented Process Modeling (POPM)

In this section we want to introduce a process modeling method which has the capability to serve as a solid basis for describing complex data integration scenarios following a structured manner. This method is called Perspective Oriented Process Modeling (POPM) [5].

POPM fosters separation of concerns in process modeling through its so called perspectives. Each perspective focuses exactly on one aspect of a process and is in general independent from any other perspective. The method itself is described in detail in [5] such that we merely want to summarize the most important features here. POPM in its basic form recognizes five main perspectives.

- The *functional perspective* defines the purpose or the goal of a process by using work steps, decisions and special marker elements that define the start and the end of a process.
- The *organizational perspective* is used for declaring which agent (human or machine) has to execute a certain element of a process (e.g. a process step). Complex rules for evaluating which individual out of an organizational structure [2] can be specified.
- The *operational perspective* contains information about what tools, services or applications can be used to carry out a certain task.
- The *data (flow) perspective* then defines producers and consumers of data inside a process by specifying which steps of a process create what data and which steps consume it again.
- The *behavioral or control flow perspective* describes causal dependencies in between two or more related tasks.

These five perspectives are the main perspectives of POPM but additional perspectives can be introduced on demand; especially when the process modeling language needs to be adapted to new application scenarios, additional perspectives can be useful (e.g. a temporal perspective which clearly defines when a certain step or task has to be started, how long it lasts and when it ends). Since POPM is based on a

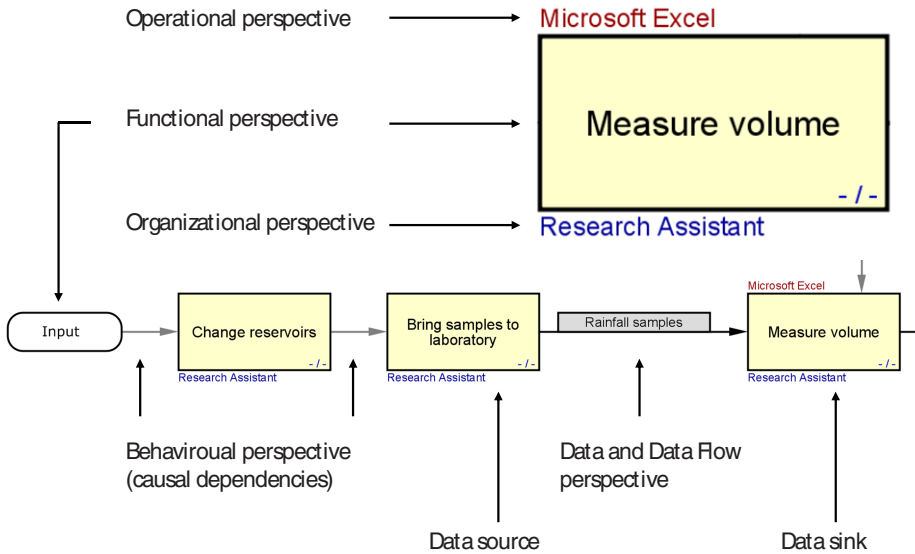


Fig. 2. An excerpt of a process modeled according to the POPM principles and showing the five basic perspectives

flexible meta model [8], these changes can be achieved easily. Fig. 2 shows an excerpt of a process stemming from ecological research [6] which demonstrates the usage and the depiction of each of the five basic perspectives.

Because of this separation of concerns, implemented with the perspectives, POPM is well structured according to the relevance of information in a process model for different groups (e.g. data experts, organizational experts etc.) which can focus their attention on those tasks they are specialized in. Another benefit of POPM is the comprehensibility of the graphical visualization – users are able to easily interpret the modeling constructs of the language such that POPM process models can be used for informal purposes; nevertheless a transformation of POPM processes into executable workflows is possible, too – [12] describes a modularized framework called the Process Driven Architecture for performing changes on the modeling language and transform process models into executable applications. Thus POPM is not only a method for process modeling but also a structured method for describing complex application scenarios and providing executable solutions for it.

DaltOn

DaltOn (“Data Logistics with Ontologies”) is a framework that provides functionality for implementing all kind of data transportation, selection, filtering and integration tasks. The interested reader may refer to [3] and [7] for a complete description of DaltOn since we can merely introduce the framework itself here.

The single actions performed by DaltOn can be depicted as a POPM process (cf. Fig. 3, only functional and behavioral perspectives shown). This process can be split into three (logical) parts: data extraction (ranging from step “Build Selection

Command” to “Perform Data Filtering”), semantic integration and data insertion (starting with the second “Transform Format” step ranging to “Insert Data”). The tasks each step has to fulfill are as follows:

- **Build Selection Command:** In order to extract data from a source, DaltOn constructs a so-called Selection Command (e.g. a SQL query) that defines the set of data to be extracted.
- **Extract Data:** This step is responsible for extracting data from the source with the help of the selection command built by the previous step. DaltOn uses wrappers as an abstraction layer for accessing different sources. As output this step delivers data in the format and semantics of the source.
- **Transform Format:** Since DaltOn cannot interpret any format available and because technically it is rather hard to introduce an abstraction of the data format, DaltOn converts data into an internal XML representation. This XML document is then being processed further.
- **Perform Data Filtering:** The task of this step is to filter out corrupted or unwanted data from the XML document. It can be specified whether the filtered data is discarded (i.e. it does not exist anymore in the document) or if the value of such an item is replaced with a pre-defined default value.
- **Semantic Integration:** Performs semantic integration based on ontologies which describe the source’s, the sink’s and the domain’s understanding of the given data. As output this step produces a new XML document which contains data following the ontology of the sink.
- **Transform Format:** After the semantic integration the XML document must be brought back into a format that is understood by the sink.
- **Build Insertion Command:** Analogously to the step Build Selection Command DaltOn will construct an insertion command that is used for storing the converted data in the right location.
- **Insert Data:** Performs the actual insert of the data using a sink specific wrapper and the insertion command.

Besides, DaltOn does not require the existence and the execution of each step explained above. For instance it is possible that no semantic transformation is needed but data only needs to be extracted from a source and is dumped into a sink (pure data transport); then DaltOn can bypass the step sequence Transform Data – Perform Data Filtering – Semantic Integration – Transform Data. It is also possible that DaltOn

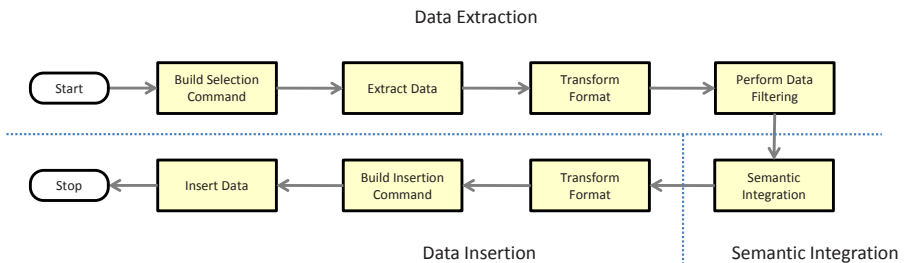


Fig. 3. The DaltOn pipeline depicted as a process

does not need to perform any action at all (e.g. if all applications use the same format and the same data store). Another feature of DaltOn is its support for data provenance; every action performed on data can be stored in a repository.

Current Integration of DaltOn in POPM

Even though DaltOn is a component which is independent of any workflow management system, it can be used for performing integration task within a workflow. Therefore efforts were made for integrating DaltOn with POPM where it became the default implementation of the data perspective. In order to ease its application, the modeling capabilities of the language were extended for supporting the configuration of DaltOn. As implied by Fig. 4, DaltOn gets called by the workflow management system every time a data related task such as data transportation or conversion is required. During modeling time the domain and application expert need to provide for each source and sink

- a format specification for data,
- a local ontology which describes how a certain application interprets data,
- a location where the source (sink) is located and
- a selection criteria for the data extraction or an insertion criteria for data insertion which is used to build the extraction and insertion commands.

Additionally a domain expert needs to provide a so-called reference ontology which is used to link the concepts of the local ontologies; this is necessary since these two ontologies might use the same concepts with different names. The reference ontology can be global for an application (i.e. a process) but it need not. Last but not least the application expert must specify criteria which are later on used in the filtering step for discarding or blanking corrupted data. What is not shown in Fig. 4 are mappings that have to be defined and specified; first, a mapping must exist which links data from the source (respectively sink) to the local ontology of the source. Second, a mapping must be defined that gives links in between a local ontology and the reference ontology.

4 Conceptual Data Integration with POPM and DaltOn

After introducing the conceptual framework for data integration and the Perspective Oriented Process Modeling methodology together with the DaltOn framework we now want to describe how the latter two can be used to provide a well structured solution for data integration scenarios. We chose POPM as a basis for the implementation since the concept of Section 0 can be nicely mapped onto the POPM perspectives.

The Technological Space so far is only contained in the data perspective which is not sufficient for our purpose. Instead of describing only a data item, also the tools which are used to perform any arbitrary action with data are to be specified in a manner such that one can recognize the format they are using. This way the data container can use a format which is more ‘durable’ than the single tools used to process data – i.e. data is recorded in a format which most likely can be still read in the future. Thus this effort is a good means for traceability of how certain results were computed. Also

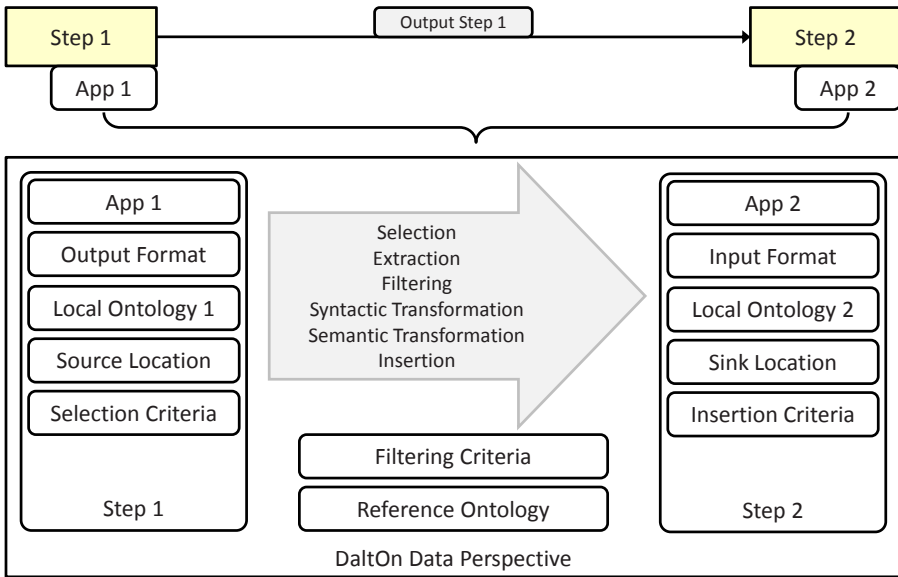


Fig. 4. Dalton is currently an extension and implementation of the POPM data perspective

agents might prefer a different format for introspecting data; for instance instead of looking at the raw data of a CSV file, an Excel sheet can be advantageous. Thus the ontologies describing the perception of an agent can be placed in the organizational perspective.

The classification of transformations in the technological space into endogenous and exogenous transformations mainly concerns Dalton only. However the design of Dalton allows both kinds of transformations since it does not convert data directly from a source format into a target format but first into an intermediate representation. Thus Dalton transformations are always exogenous transformations since it cannot be granted that source and sink format are of the same “XML” space.

Also the elements of the domain space can be mapped to POPM. The ontologies of the Analytical and Agent Spaces were called “local ontologies” in Section 0; the only difference is that now more of these ontologies exists – instead of only one per data container a process model now must define one local ontology for each tool used (operational perspective), each agent performing a task (organizational perspective) and each data item (data perspective, as explained in the last section). Additionally the Application Context contains another ontology which corresponds with the reference ontology mentioned in Section 0. Dalton as the executing instance of a data integration task is not concerned with these changes. Assigning ontologies to tools and agents makes the overall process more flexible since both can be exchanged more independently. If the information is only contained in the data perspective, the content of this perspective must always be altered too in case something changes in the process.

In order to fully support the conceptual data integration as introduced in Section 0, also DaltOn needs to be further developed; currently DaltOn uses pre-defined mappings in between the local and reference ontologies. Here techniques known from model management such as match operators [13] can help to implement an ad-hoc matching (however the result of these matchers is only a similarity value which indicates a possible match but which does not give a clear decision). But as many ontologies will contain certain similarities, the application of such operators looks promising for reducing the overhead during modeling time and for allowing the flexibility mentioned beforehand.

5 Related Work

Many scientific workflow systems support data integration. Kepler [10] and Taverna [4] are two examples. The main difference in between our approach and them is that Kepler and Taverna require specialized process steps for data integration. For performing syntactical transformations, the user of the Kepler system must introduce an actor which takes the data output of a step and puts it into the input port of the following step. For semantic data transformations the framework described in [1] can be used; similar to the DaltOn system, mappings of data into the ontologies are used (“registration mappings”) which need to be defined by the end-user as well. But unlike DaltOn, the description of concepts in the ontologies is not used for reasoning over possible matches of concepts between two ontologies. Another scientific workflow system is Taverna [4] which is a domain specific workflow management system in bioinformatics. It uses a similar approach as Kepler for integrating data; also specialized services are needed for performing syntactic and semantic transformations.

Since ontologies become more and more common in many areas, also the need for combining and aligning these ontologies emerges. [9] classifies approaches for the combination of ontologies and describes common deficiencies in this field. Basically these are mismatches in the languages used to define ontologies (different syntax, different semantics of primitives, expressivity of the language etc.) and mismatches on the ontological level (different scope of two ontologies, ontologies cover different areas of a domain, granularity of models incorrect etc.).

6 Conclusion

In this paper a conceptual approach for data integration based on process modeling and ontological descriptions of data was introduced. Instead of following the common approach of developing wrapper applications that contain all information about a specific transformation task (syntax and semantics of data) we propose to follow the separation of concerns introduced with POPM. In our approach each entity involved in an integration scenario provides the necessary information (ontologies, format descriptions etc.) itself. This way a clear separation of concerns and a low coupling between the perspectives (and thus also the implementation modules) is enforced which is a prerequisite for the implementation of conceptual data integration.

References

1. Bowers, S., Ludäscher, B.: An Ontology-Driven Framework for Data Transformation in Scientific Workflows. In: Rahm, E. (ed.) DILS 2004. LNCS (LNBI), vol. 2994, pp. 1–16. Springer, Heidelberg (2004)
2. Bussler, C.: Organisationsverwaltung in Workflow-Management-Systemen, Deutscher Universitätsverlag (in German) (1998) ISBN-10: 382442102X
3. Curé, O., Jablonski, S., Jochaud, F., Rehman, M.A., Volz, B.: Semantic Data Integration in the DaltOn System. In: Workshop on Information Integration Methods, Architectures, and Systems (IIMAS), Cancún, México (April 2008)
4. Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M.R., Li, P., Oinn, T.: Taverna: a tool for building and running workflows of services. *Nucleic Acids Res.* 34, 729–732 (2006)
5. Jablonski, S., Bussler, C.: Workflow Management – Modeling Concepts, Architecture and Implementation. Int. Thomson Computer Press, London (1996)
6. Jablonski, S., Volz, B., Rehman, M.A.: A Modeling Methodology for Scientific Processes. In: Workshop Umweltdatenbanken, Hamburg, Germany (May 2007)
7. Jablonski, S., Rehman, M.A., Volz, B., Curé, O.: Architecture of the DaltOn Data Integration System for Scientific Applications. In: Int'l. Workshop on Applications of workflows in Computational Science (AWCS), Krakow, Poland (June 2008)
8. Jablonski, S., Volz, B., Dornstaeder, S.: A Meta Modeling Framework for Domain Specific Process Management. In: 1st IEEE Int'l. Workshop on Semantics for Business Process Management (SemBPM), Turku, Finland (July 2008)
9. Klein, M.: Combining and relating ontologies: an analysis of problems and solutions. In: Proc. Of the IJCAI 2001 Workshop on Ontologies and Information Sharing, Seattle, USA, pp. 53–62 (August 2001)
10. Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger-Frank, E., Jones, M., Lee, E., Tao, J., Zhao, Y.: Scientific Workflow Management and the Kepler System. *Concurrency and Computation: Practice & Experience* 18(10), 1039–1065 (2006)
11. Mens, T., Van Gorp, P.: A Taxonomy of Model Transformation. In: Int'l. Workshop on Graph and Model Transformation (GraMoT), Tallin, Estonia (September 2005)
12. Müller, S.: Modellbasierte IT-Unterstützung von wissensintensiven Prozessen, Ph.D. Thesis, Friedrich-Alexander-University Erlangen-Nuremberg (in German) (2007)
13. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. *VLDB Journal* 10, 334–350 (2001)

Theoretical and Practical Challenges of Integrating Ecosystem Data

M. Hauhs, B. Trancón y. Widemann, and O. Archner

Ecological Modelling, BayCEER
 (Bayreuth Center of Ecology and Environmental Research)
 University of Bayreuth
 Bayreuth, Germany
 Michael.Hauhs@uni-bayreuth.de

Abstract. The challenges of data integration in the field of ecological and environmental research are exemplified by the widespread mixture of *time series* and *maps*. Typically ecosystem data sets originate from interdisciplinary studies organised around pragmatic, small scale or short-term interests. Integration of such data, e.g. in geographic information systems or time series analysis has often been based on ad hoc solutions. The theoretical basis of the field has remained largely unsolved.

Here, we propose a categorical basis for the task of organising and integrating ecosystem data. We focus on time series (data streams) and on maps (spatial configurations of objects) as the two pure limiting cases illustrating typical data evaluation tasks. The core idea of our approach is to use the duality between *algebra* and *coalgebra* as the mathematical basis for representing and relating these two data types. This results in two dual modelling paradigms that relate the formal basis to testable models and experiments in different ways.

The application of our approach to existing studies of ecosystem data provides a systematic classification of the underlying concepts (see Fig. 1). The four cases result from the algebra/coalgebra duality and from the independent logical distinction between *induction* and *deduction*. Deduction is meant in the sense that time-variable aspects of a system follow from a priority of objects. Induction, conversely, is meant in the sense that classification and control of objects of a system follow from a priority of behavior. We claim that such a classification helps to explain current difficulties and indicates new potential in ecosystem data integration.

Notions of Ecosystems	Deduction from <i>Objects</i>	Induction from <i>Behavior</i>
Non-reactive Systems as <i>Algebras</i>	Complex aggregates (species, populations, ...)	Landscape functions (hydrological runoff, ...)
Reactive Systems as <i>Coalgebras</i>	Carrier of values (endangered species, ...)	Perpetuated service (sustainable utilisation)

Fig. 1. Typical notions of ecosystems in science, management and politics

The Health Problems of Data Integration

Avigdor Gal

Technion - Israel Institute of Technology

Haifa

32000 Israel

avigal@ie.technion.ac.il

Data integration is the process of combining data residing at different data sources to generate a unified data view of these data. Schema matching generates correspondences between concepts describing the meaning of data in various heterogeneous, distributed data sources. Therefore, schema matching is recognized to be one of the basic operations required by the process of data integration and thus has a great impact on its outcome and on numerous modern applications.

We discuss the poor health of data integration and trace it back to inherent uncertainty in the schema matching process. Then, we investigate new ways to reach a common data model for schema matching. We discuss the use of matrix theory as a theoretical generic foundation to schema matching and propose to adopt the similarity matrix abstraction as a basic data model for schema matching. Such a common data model can abstract away the differences between schema matchers and focus on their similarities instead. This approach is useful in designing generic tools with wide applicability rather than solving isolated problems. Therefore, basic matching operations will be captured as matrix operations, regardless of whether the matcher itself uses a linguistic heuristic, a machine learning heuristic, *etc.* We discuss the type of abstraction that is needed for setting such a foundation and what we envision as its greatest benefits. We argue that such a framework would allow an efficient assessment of schema matcher quality, yielding a better mechanism for designing and applying new schema matchers.

Using this data model we propose a complementary approach to the design of new schema matchers. We separate schema matchers into first line and second line matchers. First line schema matchers were designed by-and-large as application of existing works in other areas (*e.g.*, machine learning and information retrieval) to schemata. Second line schema matchers operate on the outcome of other schema matchers to improve their original outcome. We aim to show the benefit in this classification, together with the use of similarity matrix as a data model, to the design of new schema matchers. Therefore, we generalize the notion of second line matchers and discuss their properties.

Given the importance of schema matching in data integration and the limitations of the state-of-the-art foundations, the significance of developing a common theoretical generic foundation for schema matching design is apparent. By having a new perspective of existing representations, we advance the state-of-the-art in providing new mechanisms to deal with the ever daunting problem of schema matching.

Computing Path Similarity Relevant to XML Schema Matching

Amar Zerdazi and Myriam Lamolle

LINC – Laboratory of Paris VIII
IUT of Montreuil – 140, rue de la Nouvelle France 93000 - Montreuil, France
{a.zerdazi,m.lamolle}@iut.univ-paris8.fr

Abstract. Similarity plays a crucial role in many research fields. Similarity serves as an organization principle by which individuals classify objects, form concepts. Similarity can be computed at different layers of abstraction: at data layer, at type layer or between the two layers (i.e. similarity between data and types). In this paper we propose an algorithm context path similarity, which captures the degree of similarity in the paths of two elements. In our approach, this similarity contributes to determine the structural similarity measure between XML schemas, in the domain of schema matching. We essentially focus on how to maximize the use of structural information to derive mappings between source and target XML schemas. For this, we adapt several existing algorithms in many fields, dynamic programming, data integration, and query answering to serve computing similarities..

Keywords: Node context, path similarity, schema matching, XML schema.

1 Introduction

This Schema matching is a schema manipulation process that takes as input two heterogeneous schemas and possibly some auxiliary information, and returns a set of dependencies, so called mappings that identify semantically related schema elements [13]. In practice, schema matching is done manually by domain experts [12], and it is time consuming and error prone. As a result, much effort has been done toward automating schema matching process. This is challenging for many fundamental reasons. According to [6], schema elements are matched based on their semantics. Semantics can be embodied within few information sources including designers, schemas, and data instances. Hence schema matching process typically relies on purely structure in schema and data instances [5]. Schemas developed for different applications are heterogeneous in nature i.e. although the data they describe are semantically similar, the structure and the employed syntax may differ significantly [1]. To resolve schematic and semantic conflicts, schema matching often relies on element names, element datatypes, structure definitions, integrity constraints, and data values. However, such clues are often unreliable and incomplete. Schema matching cannot be fully automated and thus requires user intervention, it is important that the matching process not only do as much as possible automatically but also identify when user input is necessary.

Contrary to current structural matching algorithms, we emphasize the notion of context of an element. The main goal of our works is to propose a novel approach for structural matching based on the notion of structural node context. The context of an element is given by combination of its root context, its intermediate context and its leaf context. In this paper we propose a structural algorithm that can be used for computing such context. For this, we introduce the notion of path comparison using algorithms from dynamic programming and path query answering.

The rest of paper is organized as follows. In section 2, we summarize some examples of recent schema matching algorithms that incorporate XML structural matching. Section 3 gives a brief overview of the features of XML schemas, and our formal model for XML schema (XML Schema graph). This graph is used in the matching process for the measure of node context similarity. Section 4 presents the core of this paper. We detail the different metrics necessary for computing the path similarity. After these similarities is used to determine the similarities between contexts for such elements. Section 5 concludes the paper.

2 Related Work

Schema matching is not a recent problem for the community of databases. [4] developed the ARTEMIS system employ rules that compute the similarity between schemas as a weighted sum of similarities of elements names, data types, and structural position. With the growing use of XML, several matching tools take into consideration the hierarchical and deal essentially with DTDs. In the following, we present some examples of recent schema matching algorithms that incorporate XML structural matching.

We do not present here of exhaustive manner all existing systems for schema matching, but those that appeared us interesting for the problematic that they raise or for the considered solutions.

2.1 Cupid

Cupid is a hybrid matcher combining several matching [10]. It is intended to be generic across data models and has been applied to XML and relational data sources. Cupid is based on schema comparison without the use of instances. Despite these extensions, Cupid does not exploit all XML schema features such as substitution groups, abstract types, etc that could give a significant clue in solving XML schema matching problem.

2.2 LSD

The LSD (Learning Source Description) system [5] uses machine-learning techniques to match a new data source against a previously defined global schema. LSD is based on the combination of several match result obtained by independent learners. This approach presents several limitations since it does not fully exploit XML structure. Besides, the only structural relationship considered within the LSD system is the parent-child relationship, which is not sufficient to describe the context of elements to matcher.

2.3 Similarity Flooding

In [11], authors present a structure matching algorithm called Similarity Flooding (SF). The SF algorithm is implemented as part of a generic schema manipulation tool that supports, in addition to structural SF matcher, a name matcher, schema converters and a number of filters of choosing the best match candidates from the list of ranked map pairs returned by the SF algorithm. SF ignores all type of constraints while performing structural matching. Constraints like typing and integrity constraints are used at the end of the process to filter mapping pairs with the help of user.

2.4 SemInt

SemInt [8], [9] represents a hybrid approach exploiting both schema and instance information to identify corresponding attributes between relational schemas. The schema-level constraints, such as data type and key constraints are derived from the DBMS catalog. Instance data are exploited to obtain further information, such as actual value distributions, numerical averages, etc. For each attribute, SemInt determines a signature consisting of values in the interval $[0,1]$ for all involved matching criteria. The signatures are used first to cluster similar attributes from the first schema and then to find the best matching cluster for attributes from the second schema. The clustering and classification process is performed using neural networks with an automatic training, hereby limiting pre-match effort. The match result consists of clusters of similar attributes from both input schemas, leading to m:n local and global match cardinality.

3 Data Model

As we already mention in section 2, up to now few existent XML schema matching algorithms focus on structural matching exploiting all W3C XML schemas [14] features. In this section, we propose an abstract model that serves as a foundation to represent conceptually W3C XML schemas and potentially other schema languages. We model XML schemas as a directed labelled graph with constraint sets; so-called schema graph. Schema graph consists of series of nodes that are connected to each other through directed labelled links. In addition, constraints can be defined over nodes and links. In [15], we detail the proposed model for XML schemas in order to define a formal framework for solving matching problem. Figure 1 illustrates a schema graph example.

3.1 Features of XML Schema

The XML schema language incorporates the following features.

The structure of an XML document is defined in an XML schema in terms of pre-defined hierarchical relationships between XML elements and/or attributes to which specific constraints concerning ordering, cardinality and participation are imposed (e.g., `xs:element`, `xs:attribute`, `xs:sequence`, `xs:all`, `xs:choice`, `xs:minOccurs`, `xs:use`, etc.).

The content of an XML document as found in elements or attributes can be restricted in an XML schema by defining it to take values from a domain of a predefined or user-defined datatype (e.g., `xs:string`, `xs:simpleType`, `xs:restriction`, `xs:union`, etc.).

Semantic invariants can be enforced in XML schema by imposing referential integrity or uniqueness constraints (e.g., `xs:key`, `xs:keyref`, `xs:unique`, etc.).

Features supporting modularity and reusability in XML schema enable rapid schema development and reuse of, possibly adjusted, predefined schemas (e.g., `xs:import`, `xs:include`, `xs:group`, `xs:extension`, etc.).

Finally, documentation features facilitate human and machine understanding of an XML schema (e.g., `xs:annotation`, `xs:documentation`, etc.).

3.2 XML Schema Graph

Schema graph nodes

We categorize nodes into atomic nodes and complex nodes. Atomic nodes have no edges emanating from them. They are the leaf nodes in the schema graph. Complex nodes are the internal nodes in the schema graph. Each atomic node has a simple content, which is either an atomic value from the domain of basic data types (e.g., string, integer, date, etc.). The content of a complex node, called complex content, refers to some other nodes through directed labeled edges. In figure 1, nodes `laboratory` and `library` are complex nodes, while nodes `name` and `location` are atomic nodes.

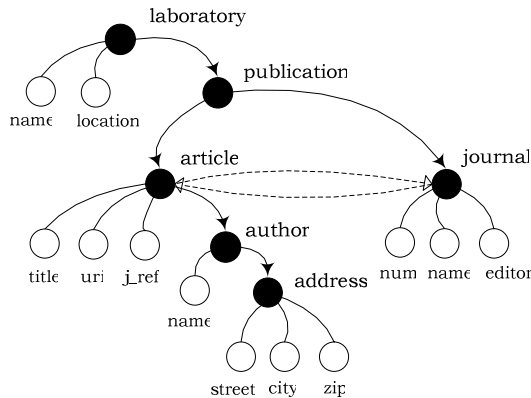


Fig. 1. An EXS schema graph example

Schema graph edges

Each edge in the schema graph links two nodes capturing the structural aspects of XML schemas. We distinguish two kinds of edges: (i) implicit edges (e.g. the parent/child relationships between elements), they are depicted with a solid line edges. (ii) explicit edges defined in XML schema by means of `xs:key` and `xs:keyref` pairs or similar mechanisms. They are represented using a pair of reverse parallel edges (generally bidirectional, specifying that both nodes are conceptually at the same level:

association relationship). In figure 1, an implicit edge links the two nodes laboratory and library. An explicit edge between journal and article specifies a key/keyref relation.

Schema graph constraints

Different constraints can be specified with XML Schema language. These constraints can be defined over both nodes and edges. Typical constraints over an edge are cardinality constraints. Cardinality constraints over a containment edge specify the cardinality of a child with respect to its parent. Cardinality constraints over an implicit edge imply for example an optional or mandatory attribute for a given node. The default cardinality specification is [1,1]. We also distinguish three kinds of constraints over a set of edges: (i) ordered composition, defined for a set of containment relationships and used for modelling XML Schema “sequences” and all mechanisms; (ii) exclusive disjunction, used for modelling the XML Schema choice and applied to containment edges; and (iii) referential constraint, used to model XML schema referential constraints. Referential constraints are applied to association edges. Other constraints are furthermore defined over nodes. Examples include uniqueness and domain constraints. Domain constraints are very broad. They essentially concern the content of atomic nodes. They can restrict the legal range of numerical values by giving the maximal/minimal values; limit the length of string values, or constrain the patterns of string values.

Node Context definition

Our aim of structural matching is the comparison of the structural contexts in which nodes in the schema graph appear. Thus, we need a precise definition on what we mean by node context. We distinguish three kinds of node contexts depending on its position in the schema graph:

The root-context: of a node n is defined as the path (going through containment edges) having n as its ending node and the root of the schema graph as its starting node. Example, the root-context of node publication in figure 1 is given by the path node laboratory/publication indicating that the node publication describes the publications belonging to a laboratory. The ancestor-context of the root node is empty and it is assigned a null value.

The intermediate-context: of a node n includes its attributes and its immediate subelements. The intermediate-context of a node reflects its basic structure and its local composition. The intermediate-context of an atomic node is assigned a null value. Example, the intermediate-context of node publication in the schema graph of figure 1 is given by (article, journal). The intermediate-context of an atomic node is assigned a null value.

The leaf-context: leaves XML documents represent the atomic data that the document describes. The leaf-context of a node n includes the leaves of the subtrees rooted at n . Example; the leaf-context of node publication in the schema graph of figure 1 is given by (street, city, zip, num, name, editor). The leaf-context of an atomic node is assigned a null value.

The context of a node is defined as the union of its root-context, its intermediate-context and its leaf-context. Two nodes are structurally similar if they have similar

contexts. To measure the structural similarity between two nodes, we compute respectively the similarity of their root, intermediate and leaf contexts [16]. The notion of context similarity has been used in Cupid and SF; however none of them relies on the three kinds of contexts. To measure the structural similarity between two nodes, we compute respectively the similarity of their root, intermediate and leaf contexts. In the following we describe the basis needed to compute such similarity.

4 Path Similarity Measure

Structural node context defined in the previous section relies on the notion of *path*. In order to compare two contexts, we essentially need to compare two paths. Path comparison has been widely used in answering conjunctive queries.

Let us consider two paths $ph_1(G_1, sequence_1) = \langle G_1, n_{i_1}, \dots, n_{i_m} \rangle$ and $ph_2(G_2, sequence_2) = \langle G_2, n_{j_2}, \dots, n_{j_l} \rangle$. A mapping between ph_1 and ph_2 is an assignment function $\varphi: ph_1 \rightarrow ph_2$ that associates a node in ph_1 to a node in ph_2 . An assignment φ is a strong matching if it satisfies the two following conditions:

- *Root constraint*: Source nodes in ph_1 and ph_2 are similar. Two nodes are considered similar if their similarity exceeds a specified threshold with respect to a predefined function.

- *Edge constraint*: For directed edge $\mu \rightarrow v$, where $\mu, v \in \langle G_1, n_{i_1}, \dots, n_{i_m} \rangle$, there exist directed edge $\mu' \rightarrow v'$, where $\mu', v' \in \langle G_2, n_{j_2}, \dots, n_{j_l} \rangle$ such that nodes μ, μ' are similar nodes and v, v' are similar nodes.

The definition of strong matching reminds us the classical view of a conjunctive query and an answer to it. Under such conditions paths such as *author/publication* and *publication/author* are not matched however they convey same semantics. Other unmatched paths under such conditions are *author/contact/address* and *author/address*. Based on such observations, it is more appropriate to go beyond the strong matching by relaxing the above conditions. One can think of several ways of relaxing strong matching: for example allow matching paths even when nodes are not embedded in a same manner or in the same order. Several works in query answering have proposed relaxation issues to approximate answering of queries (including path queries) [2]. Inspired by [3] work in answering XML queries we made the following relaxations:

- *Root constraint relaxation*: Paths can be matched even if their source nodes do not match, for example *author/publication* may match *staff/authors/author/publication*.

- *Edge constraint relaxation*: Paths can be matched even if their nodes appear in different order *author/publication* and *publication/author*. Paths can also be matched even if there are additional nodes within the path (e.g. *author/contact/address* match *author/address*) meaning that the *child-parent* edge constraint is relaxed into *ancestor-child* constraint.

Relaxations may give rise to multiple match candidates. For this reason, authors in [3] define a path resemblance measure between a given path query Q and a path in the source tree. Such measure is used for ranking match candidates. We extend these definitions by allowing two elements within each path to be matched, even if they are not identical but their linguistic similarity exceeds a fixed threshold. We define a *path*

resemblance measure, denoted pr , which determines the similarity between two given paths. The values of $phSim$ range between 0 and 1. Match candidates can then be ranked according to pr measure. Consider two paths ph_1 and ph_2 being matched (when ph_1 is a target path and ph_2 is a source path), ph_2 is the best match candidate for ph_1 if it fulfills the following criteria:

- The path ph_2 includes most of the nodes of ph_1 in the right order.
- The occurrences of the ph_1 nodes are closer to the beginning of ph_2 than to the tail, meaning that the optimal matching corresponds to the leftmost alignment.
- The occurrences of the ph_1 nodes in ph_2 are close to each other, which mean that the minimums of intermediate non-matched nodes in ph_2 are desired.
- If several match candidates that match exactly the same nodes in ph_1 exist, ph_2 is the shortest one.

To calculate $phSim(ph_1, ph_2)$, we first represent each path as a set of string elements; each element represents a node name (e.g., the path *Author/Publication* is a string composed two string elements *Author* and *Publication*). We used the four scores established in [3] and borrowed from dynamic programming for string comparison; each of which corresponds to one of the above criteria.

4.1 Longest Common Subsequence

To answer the first criterion, we use a classical dynamic programming algorithm in order to compute the *Longest Common Subsequence (LCS)* [7], between ph_1 and ph_2 . More the length of the longest common subsequence is high; more ph_2 includes ph_1 nodes in the right order.

A word w is a longest common subsequence of x and y if w is a subsequence of x , a subsequence of y and its length is maximal. Two words x and y can have several different longest common subsequences. The set of the longest common subsequences of x and y is denoted by $LCS(x, y)$. The (unique) length of the elements of $LCS(x, y)$ is denoted by $lcs(x, y)$. For comparing two words x and y of size m and n respectively, we reuse a classical dynamic programming algorithm that relies on two-dimensional table $T[0..m, 0..n]$. We then exhibit the longest common subsequence tracing back in table from $T[m-1, n-1]$ to $T[-1, -1]$. Finally, to obtain a score in $[0, 1]$, we normalize the length of the longest common subsequence by the length of target path ph_1 as following:

$$lcs_n(ph_1, ph_2) = |lcs(ph_1, ph_2)| / |ph_1|$$

Example. Consider ph_1 to be *publication/book/author* and ph_2 as *author/publication/book*, the longest common subsequence between the two paths as *publication/book*, $lcs(ph_1, ph_2) = 2$, thus $lcs_n = 2/3 = 0.66$.

4.2 Average Positioning

To answer the second criterion, we first compute, according to $lcs(ph_1, ph_2)$ what would be the average positioning of the optimal matching of ph_1 within ph_2 . The optimal matching is the match that starts on the first element of ph_1 and continues without gaps. Consider $ph_1 = \textit{author/publication/book}$ and $ph_2 = \textit{staff/author/publication/book}$, since the optimal matching corresponds to the leftmost alignment, the average optimal position,

denoted $optPos$ is $(1+2+3)/3 = 2$. We then evaluate using the LCS algorithm, the actual average positioning ($avgPos$). $avgPos$ takes the value 3 in our example $((2+3+4)/3)$. Last, we compute pos coefficient indicating how far the actual positioning is from the optimal one, using the following formula:

$$pos(ph_1, ph_2) = 1 - [(avgPos - optPos) / (|ph_2| - 2 \times optPos + 1)]$$

4.3 LCS with Minimum Gaps

To answer the third criterion, we use another version of the LCS algorithm in order to capture the LCS alignment with minimum gaps. If $ph_1=person/address$ and $ph_2=person/contact/address$, we count a gap of length 1 between the two paths, thus $g = 1$. To ensure that we obtain a score inferior to 1, we normalize the obtained gap using the following formula:

$$gap(ph_1, ph_2) = g / (g + lcs(ph_1, ph_2))$$

4.4 Length Difference

Finally, in order to give higher values to source paths whose length is similar to the target path, we suggest to compute the length difference ld between a source path ph_1 and $lcs(ph_1, ph_2)$ normalized by the length of ph_1 as follow:

$$ld(ph_1, ph_2) = (|ph_2| - lcs(ph_1, ph_2)) / |ph_1|$$

To obtain the path similarity score, all the above metrics are combined as follow:

$$phSim(ph_1, ph_2) = \alpha lcs_n(ph_1, ph_2) + \beta pos(ph_1, ph_2) - \lambda gap(ph_1, ph_2) - \delta ld(ph_1, ph_2)$$

Where α , β , λ and δ are positive parameters ranging between 0 and 1 that represent the comparative importance of each factor. They can be tuned but must satisfy $\alpha + \beta = 1$, so that $phSim(ph_1, ph_2) = 1$ in case of a perfect match, and λ and δ must be chosen small enough so that pr cannot take a negative value. The following algorithm summarizes the computation of path similarity measure using the above formulas.

1. Input: $ph_1, ph_2, \alpha, \beta, \lambda, \delta$
2. Output: $phSim(ph_1, ph_2)$
3. Begin
4. //score 1: computation of the longest common subsequence
5. $lcs(ph_1, ph_2) \leftarrow \text{TRACE-BACK}(\text{LSC}(ph_1, ph_2))$
6. $lcs_n(ph_1, ph_2) \leftarrow lcs(ph_1, ph_2) / |ph_2|$
7. //score 2: computation of average positioning
8. $pos(ph_1, ph_2) = 1 - [(avgPos - optPos) / (|ph_2| - 2 \times optPos + 1)]$
9. //score 3: computation of LCS with minimum gaps
10. $gap(ph_1, ph_2) = g / (g + lcs(ph_1, ph_2))$
11. //score 4: computation of length difference
12. $ld(ph_1, ph_2) = (|ph_2| - lsc(ph_1, ph_2)) / |ph_2|$
13. //computation of path similarity
14. $phSim(ph_1, ph_2) = \alpha lcs_n(ph_1, ph_2) + \beta pos(ph_1, ph_2) - \lambda gap(ph_1, ph_2) - \delta ld(ph_1, ph_2)$
15. return $phSim$
16. Fin.

Example. Let $ph_1 = \text{laboratory/ author/ publication/book/description/title/subtitle}$ and $ph_2 = \text{author/book/title}$.

We have $lcs(ph_1, ph_2) = (2+3+4)/3 = 3$,

$avgPos = (2+4+6)/3 = 4$, $g = 2$, and $ld = 7-3/7 = 4/7$.

Taking α , β , λ and δ and d to respectively 0.75 , 0.25 , 0.25 , 0.2 . Note though that more extensive experimentation is needed to decide on the ideal parameters. We obtain a path similarity score equal to 0.68 .

5 Conclusion

In this paper we have interested on schema matching, and focused on the notion of path context for comparing the structural context similarity. The context element in our approach is given by the combination of three structural contexts.

We began by an analysis of problems involved in the matching, and we proposed a new solution taking into account of heterogeneity of the schema sources. We outlined the limitations of current solutions through the study of Cupid and Similarity Flooding systems and SemInt. Then we proposed a structural matching technique that considers the context of schemas nodes (defined by their roots, intermediates and leafs contexts in schema graph). By the way, we suggest a simple algorithm based on the previous ideas and exploit the three types of contexts for capturing the similarity between elements of schema graph. For this we combine a classical dynamic programming algorithm and four scores established: The longest common subsequence, the average positioning, LCS with gaps and length difference to serves computing this path similarity measure.

For future work, we would like to improve the matching process, while taking into account the optimisation of the process in order to determine a set of semantic equivalences between schemas (source and target). That will facilitate the generation of operators based on the primitive of transformations between elements of XML schemas.

References

1. Abiteboul, S., Cluet, S., Milo, T.: Correspondence and Translation for heterogeneous data. In: Afrati, F.N., Kolaitis, P.G. (eds.) ICDT 1997. LNCS, vol. 1186, pp. 351–363. Springer, Heidelberg (1996)
2. Amer-Yahia, A., Cho, S., Srivastava, D.: Tree Pattern Relaxation. In: Proceedings of DBT 2002 (2002)
3. Carmel, D., Efraty, G., Landau, G.M., Maarek, Y.S., Mass, Y.: An Extension of the vector space model for querying XML documents via XML fragments. In: XML and IR Workshop, 2nd edn., SIGIR Forum (2002)
4. Castano, S., De Antonellis, V.: A schema analysis and Reconciliation Tool Environment For Heterogeneous Databases. In: Proceedings of International Database Engineering and Applications Symposium (1999)
5. Doan, A., Madhavan, J., Domingos, P., Halevey, A.: Reconciling schemas of disparate data sources: A machine Learning Approach. In: Proceedings ACM SIGMOD conference, pp. 509–520 (2001)

6. Drew, P., King, R., McLeod, D., Rusinkiewicz, M., Silberschatz, A.: Report of the Workshop on Semantic Heterogeneity and Interoperation in Multidatabase Systems. In: Proceedings ACM SIGMOD record, pp. 47–56 (1993)
7. Hirschberg, D.S.: A Linear Space Algorithm for Computing Maximal Common Subsequences. Communications of the ACM (1975)
8. Li, W.S., Clifton, C.: Semantic Integration in Heterogeneous Databases Using Neural Networks. VLDB (1994)
9. Li, W.S., Clifton, C.: SemInt: A Tool for Identifying Attribute Correspondences in Heterogeneous Databases Using Neural Network. Data and Knowledge Engineering (2000)
10. Madhavan, J., Bernstein, P., Rahm, E.: Generic schema matching with cupid. VLDB (2001)
11. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity Flooding: A versatile Graph Matching and its Application to Schema Matching. Data Engineering (2002)
12. Miller, A.G., Hass, L., Hernandez, M.A.: Schema mapping as query discovery. VLDB, 77–88 (2000)
13. Rahm, E., Bernstein, P.: A survey of approaches to automatic schema matching. VLDB Journal, 334–350 (2001)
14. XML Schema, W3C Recommendation, XML-Schema Primer, W3 Consortium (2001), <http://www.w3.org/TR/xmlschema-0>
15. Zerdazi, A., Lamolle, M.: Modélisation des schémas XML par adjonction de métaconnaissances sémantiques. In: ASTI 2005 (2005)
16. Zerdazi, A., Lamolle, M.: Matching of Enhanced XML Schema with a measure of structural-context similarity. In: Proceeding of The 3rd International Conference on Web Information Systems and Technologies, WEBIST 2007 (2007)

Improving Search and Navigation by Combining Ontologies and Social Tags

Silvia Bindelli**, Claudio Criscione**, Carlo A. Curino*, Mauro L. Drago**,
Davide Eynard*, and Giorgio Orsi*

Dipartimento di Elettronica e Informazione, Politecnico di Milano,
via Ponzio 34/5, 20133 Milano, Italy

* {Curino,Eynard,Orsi}@elet.polimi.it

** {Silvia.Bindelli,Claudio.Criscione,MauroL.Drago}@gmail.com

Abstract. The Semantic Web has the ambitious goal of enabling complex autonomous applications to reason on a machine-processable version of the World Wide Web. This, however, would require a coordinated effort not easily achievable in practice. On the other hand, spontaneous communities, based on social tagging, recently achieved noticeable consensus and diffusion. The goal of the *TagOnto* system is to bridge between these two realities by automatically mapping (social) tags to more structured domain ontologies, thus, providing assistive, navigational features typical of the Semantic Web. These novel searching and navigational capabilities are complementary to more traditional search engine functionalities. The system, and its intuitive AJAX interface, are released and demonstrated on-line.

Keywords: ontology, tag, web2.0, social bookmarking, search engine.

1 Introduction

The Semantic Web is the “high road” toward a better exploitation of the vast amount of heterogeneous data available in the web. The overall goal is to mediate the access to existing sources, by means of formalized, shared, and explicit representation of the data semantics through ontologies, and to deliver value added interactions. This “high road”, appreciated in the academic environments, requires high switching costs and a wide distributed and coordinated effort, which is hard to achieve in practice. On the other hand, the recent phenomenon of the Social Web and in particular of tag-based systems represents a more practical and viable “low road” toward a better fruition of the web. The goal of the *TagOnto* system is to bridge the two roads, by automatically mapping tag-based systems with the more structured world of ontologies. The main contribution of our approach is to enhance the user experience by providing features typical of the “high road” while requiring only limited commitment, typical of the “low road”, from users and content providers. The system exploits a rich set of heuristics, ranging from simple string-distance measures to web-based tag disambiguation techniques, to discover correspondences between tags and concepts of domain ontologies.

Therefore, the unstructured and uncontrolled nature of the *folksonomies*—as often the social tagging systems are named—is balanced by the formal rigor of the ontology-based component of our system. *TagOnto* enriches the user browsing experience by enhancing navigation and tag-based search with ontology-based search capability, which allows to disambiguate tags and to focus the user attention. The system platform is available for download and testable as an on-line demo¹. Both in the demo and in the paper we use the simple and well-known Wine ontology² as a running example. To show system extensibility we integrate in this example not only the standard tag engines such as *del.icio.us*, but also the wine community *Vimorati*.

The paper is organized as follows: Section 2 provides a brief overview of the system functionalities, Section 3 summarizes background knowledge, Section 4 discusses the internals of the system from a conceptual point of view, while architectural aspects are discussed in Section 5. Section 6 presents some related works, and Section 7 draws our conclusions.

2 System Overview

TagOnto is a *folksonomy aggregator* that offers services to relate, navigate and combine results of different tag-based systems. The key features of the system are: a *tag-based search engine*, mashing up several folksonomies to retrieve resources (bookmarks, images and videos); an *ontology-based query refinement*, exploiting a domain ontology, co-occurrence of tags and disambiguation techniques to filter prior results; and an *ontology-based navigation interface*, allowing the user to retrieve further results by graphical navigation of the ontology concepts. The above features provide two orthogonal and complementary ways, typical respectively of social and semantic web, to navigate the search results: associated-tag and ontology-navigation. The ontology is used as a common vocabulary and bridges the various folksonomies integrated in the system as a global schema of

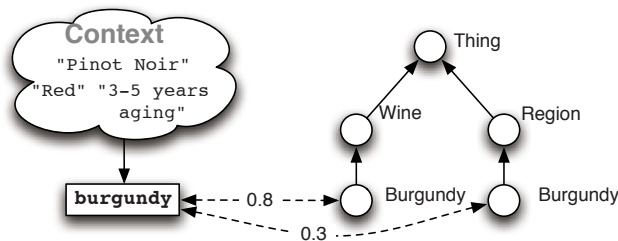


Fig. 1. An example of tag to ontology matching

¹ The on-line demo can be reached from: <http://kid.dei.polimi.it/tagonto>

² Available at: <http://www.w3.org/TR/2003/CR-owl-guide-20030818/wine>

a federated database; the system provides facilities to efficiently load the desired ontology before starting a web search. The typical user interaction is the following: (i) the user searches for a tag, e.g., *burgundy* (see Figure 1), (ii) navigates the concept of the associated ontology to refine the query, e.g., by selecting *burgundy* as a *wine* instead of as a *region*, or (iii) makes the query more general by navigating on more abstract concepts in the ontology. These actions are intuitively supported by the AJAX interface discussed in Section 5.

The associations between tags and ontology concepts are automatically discovered by the system, but also added, improved and maintained collaboratively. The automatic discovery of associations between folksonomies and domain ontology, represented as dashed lines in Figure 1, is based on a set of matching algorithms computing similarities. Disambiguation heuristics are then used in order to debug multiple associations between tags and concepts in the ontology.

Folksonomies are accessed by using dedicated wrappers exploiting three main methods to retrieve the needed resources: (i) Web2.0 APIs, (ii) RSS feeds, and (iii) Page scraping. The first approach, by relying on existing APIs offered by Web2.0-enabled websites, is our preferred one. In the second approach, the source of information is an RSS feed parsed and processed by a dedicated wrapper. The last technique is used when no other solutions are available; *TagOnto* uses page scraping to retrieve the needed information by making extensive use of regular expressions over the webpages to obtain tags and resources associated with them.

3 Background

We now introduce key background notions used throughout the paper:

Ontology: an ontology, according to T. Gruber [13], defines a set of representational primitives which can model a domain of knowledge or discourse. We can formally define an ontology as a 4-tuple $O = \langle \mathcal{C}, \mathcal{R}, \mathcal{I}, \mathcal{A} \rangle$ where \mathcal{C} is a set of *concepts* (or classes) which are subsets of a common domain Δ , \mathcal{R} is a set of relations including both binary relations between classes, called *roles* and binary relations between concepts and datatypes, called *attributes*, \mathcal{I} is a set of individuals (or ground symbols) which belong to Δ , and \mathcal{A} is a set of axioms (or assertions) in a logical form which are valid in the domain and restrict the number of possible interpretations of the ontological model.

Folksonomy: folksonomies are commonly defined as *the result of personal free tagging of information and resources for one's own retrieval* [17]. A *tag* in *TagOnto* is represented as a pair $T = \langle t, u \rangle$ where t is a term and u is a web resource (i.e., URL, image or video). The tagging is done in an open (social) environment, thus, the system is generated from the tagging performed by the people, which act both as tag providers and consumers. The term *folksonomy* derives from *folk* (people) and *taxonomy*. This is, however, often misleading since folksonomies lack the structure typical of taxonomies.

4 Matching and Disambiguation

As sketched in Section II, one of the main problems in *TagOnto* is how to match a tag to a concept in the ontology. Given a tag and a reference domain ontology, the *matching process* (i) searches the ontology for *named concepts* whose name matches the tag, and (ii) looks for *related terms* which may refine the query for a better search. Moreover, (iii) a disambiguation process is often needed to reduce the noise produced by the collaborative tagging. Once the association has been created, the matched concepts are associated to each resource tagged by the corresponding tags. More precisely, given the set \mathcal{T} of all the available tags and the set \mathcal{C} of all the named concepts defined in a specific ontology, the matching is defined as a relation $M \subseteq \mathcal{T} \times \mathcal{C}$. The relation M allows multiple associations between tags and concepts. Figure 1 shows an example of such ambiguity: the term *Burgundy* might be referred either to the wine with that specific appellation or the region of France where that particular wine is produced. To distinguish the two different word acceptations, *TagOnto* associates to each matching a similarity degree by introducing the function $s : \mathcal{T} \times \mathcal{C} \rightarrow [0, 1]$.

To establish the matchings and to compute the similarity degree, *TagOnto* relies on the set of matching algorithms shown in Table 1. The matching algorithms can be classified on the basis of their effect on the set of matchings, in particular we distinguish between *generators* which generate new matchings starting from a tag and previous matchings and *filters* which choose the best candidates from a set of matchings. Another classification considers the metrics used to compute the matching degrees; we can distinguish between *language-based matching* which uses only morphological and lexical information such as string-distance metrics to compute the similarity and *semantic matching* which uses semantic and background knowledge to create new matchings. Notice that the matching problem has been extensively studied for ontologies [6] and many different classifications are present in the literature. In our context, the main difference is the absence of structure in folksonomies which does not allow an exploitation of structural similarities between the terms in the folksonomy and those in the ontology. Language-based generators use well known string-distance metrics, such as Jaccard similarity and Levenshtein distance. On the contrary, an example of language-based filter is the *Google Noise* algorithm, which suggests possible corrections for misspelled keyword by using the “did you mean”

Table 1. Some matching heuristics

	Language-based	Semantic
Generators	Levenshtein Distance Jaccard Similarity Google Noise Correction Concept Instances Similarity	Wordnet Similarity
Filters	Max Threshold	Graph Connectivity Neighbors Google Search

feature of Google. In a similar way, a semantic generator is the *WordNet Similarity* algorithm which computes the Leacock-Chodorow [12] distance metric in WordNet between the term used in the tag and the concepts of the ontology. In *TagOnto* we use the implementation of the algorithm which is used in X-SOM (eXtensible Smart Ontology Mapper) [2] since it offers some extensions to handle compound words, acronyms and language conventions which are quite common in both folksonomies and ontologies. Since *TagOnto* is supposed to work online and with a fast response time, the class of syntactic filters includes some rather simple algorithms to select the best candidate matchings for a given tag, some examples are the *threshold filter*, which selects only matchings having a similarity degree greater than a specified threshold, and the *max filter* which selects the k matchings with the highest similarity degree. On the contrary, semantic filters are extremely useful in the disambiguation process since they alter the similarity degree of a matching by analyzing the concepts correlated to a tag using the structural information of the ontology. The disambiguation process is composed of two steps: (i) given a tag, the most frequent co-occurring tags are retrieved in order to specify its meaning (i.e., its *context*), and (ii) the ontology is analyzed in order to identify the concept which the closest meaning to the tag in that particular context.

The first process is carried out by the *Google filter* algorithm which retrieves the co-occurrent tags by issuing a query into Google and analyzing the first result. The second step, called *Neighbors filtering* leverages a common functionality of tag-based systems: the *tag-clouds*, which associate to each tag another set of tags whose meaning is correlated to the original one. After this information has been retrieved, *TagOnto* updates the similarity degrees of the matchings. As an example (see Figure 2) suppose we have the tag *Burgundy* with multiple matching concepts in the ontology (called *root concepts*); in first place *TagOnto* matches the co-occurrent tags obtained from tag clouds with the concepts of the ontology. The second step leverages the structure of the ontology by counting, for each matching, the number of links which connect matched concepts with each root concept, producing a vector of connectivity degrees v . The last step modifies the matching degrees of the root concepts according to the connectivity degrees computed in the previous step. For each matching i , *TagOnto* computes an offset measure $\varepsilon_i = \frac{D[i]}{MAX(v)}$ which is compared with the average connectivity $AVG(v)$; if $\varepsilon_i < AVG(v)$ then the new matching degree is decreased by a factor $\alpha \cdot \varepsilon_i$ where $\alpha \in [0, 1]$ is a configurable discount factor (currently set to 0.2 after the test phase); in the same way, the matching degree is increased if $\varepsilon_i > AVG(v)$. If the updated matching degree exceeds the values in $[0, 1]$ the value is truncated to fit the range.

How these heuristics are combined depends on the selected matching strategy. We provide two different strategies: a *greedy strategy* which first invokes the syntactic and semantic generators and then applies the syntactic filters, and the *standard strategy* which invokes the greedy strategy and then disambiguates the results by invoking semantic filters. When tagging occurs in small communities of practice, which share a specific vocabulary without many ambiguities, the

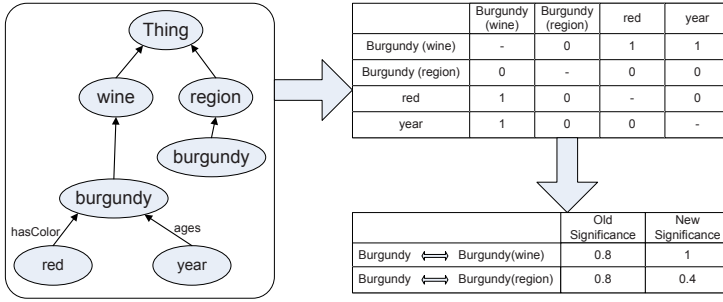


Fig. 2. An example of the disambiguation process

greedy strategy can provide results comparable with the standard one, but in a shorter time. Whenever, instead, the user base is large and heterogenous such as on the Internet, the higher cost of semantic disambiguation is compensated with a much higher quality.

5 Architecture

The overall architecture of *TagOnto* is logically divided into three different components: a tag-based search engine extensible with plugins, a heuristic matching discovery engine and a web-based user interface.

TagontoNET: TagontoNET provides core search engine functionalities and takes care of the integration of the results coming from folksonomies. The plugin-based architecture decouples the interaction between tag providers and *TagOnto*'s business logic. The system currently implements seven plugins to interact with some of the most popular tag-enabled websites such as Flickr, YouTube, del.icio.us, and Zvents. TagontoNET offers two main functionalities: tag-based resource retrieval and neighboring tag computation (needed by TagontoLib as discussed in the following). The results are delivered through a RESTful [7] web service, implemented in PHP, to further decouple this functionality, which might be used independently with the ontology-based portion of *TagOnto*.

TagontoLib: a Java library implementing the core matching functionalities of the system. The matching engine developed in Java implements the matching heuristics and strategies described in Section 4. To overcome performance limitations an effective caching technique has been built, maintaining recent matching tags and ontological concepts. As for the previous component, much attention has been devoted to the modularization of the tool. The communications between this library and the interface has been, in fact, based on a REST-like communication paradigm [7].

TagOnto Web Interface: one of the distinguishing features of *TagOnto* is its web Interface which offers to the user the support of the Ontology within a comprehensive view of the results collected from a number of different tag engines. Users can import new ontologies into the system just by entering their URIs into a special page. The interface is then divided into two horizontal portions: the upper one reports the search results, the lower one is dedicated to the ontology representation and navigation. Each user query triggers searches in both the ontology and the tag-engines. The results from these two sources are respectively shown in the upper and in the lower part of the page. This provides a unified view of the ontological meaning of a tag and the available resources (tagged with that keyword). It is possible to exploit the support of the ontology to improve the search by navigating the ontology and thus triggering a query refinement procedure that will retrieve more specific resources based on the associated tags.

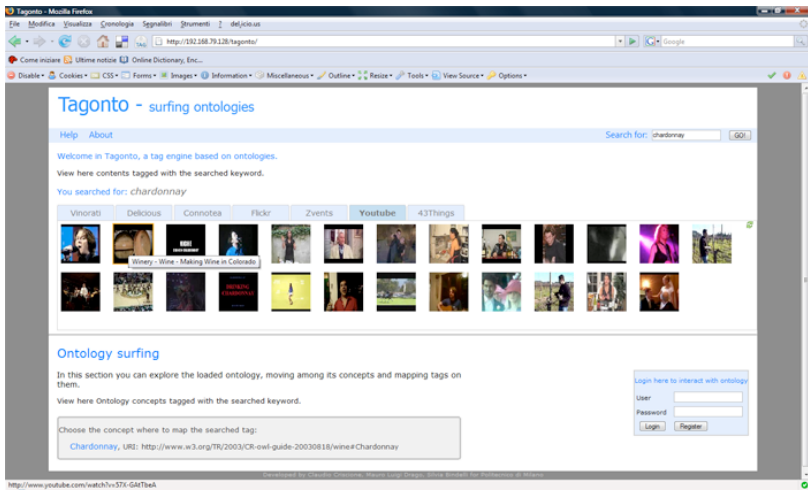


Fig. 3. The basic *TagOnto* web interface

The interface provides several tabs reporting the results obtained by searching each folksonomy. Textual results are presented in a “Google-like” way, while for picture results (e.g., Flickr resources) a thumbnail of the matching image is shown. The lower part of the page is dedicated to the presentation of the ontological concepts associated to the search. When a keyword is typed in the search field, a so-called “disambiguation box” appears in this area, to let the user choose among the concepts *TagOnto* computes as best matches. Once a concept has been chosen, previously mapped tags and resources are shown. The system also provides a box-based representation of other concepts related to the selected one, allowing an ontology-based navigation. During this navigation process the co-occurrence of tags is used to provide feedback to the user and to suggest further directions for the exploration.

5.1 Performance

We measure system performance in terms of efficiency of the analysis and matching process, while an extensive usability study is part of our research agenda. To measure system efficiency, we stress test *TagOnto* when performing the two most expensive tasks occurring at run-time: (i) the time needed by *TagOnto* to analyze a new ontology to be deployed, and (ii) the time needed to automatically generate matchings. Figure 4 shows outcomes of our analysis. The time needed to perform an ontology analysis depends mostly on the number of concepts and properties declared in the ontology, with polynomial complexity as shown in Figure 4(a) while, with fixed concepts and properties (i.e., fixed schema), the number of instances declared in the ontology influences the execution time linearly as shown in Figure 4(b). Figure 4(c) shows the distribution of response time obtained by issuing 344 tag-queries (i.e., queries composed by a single term) taken from a set of terms referring to the wine domain.

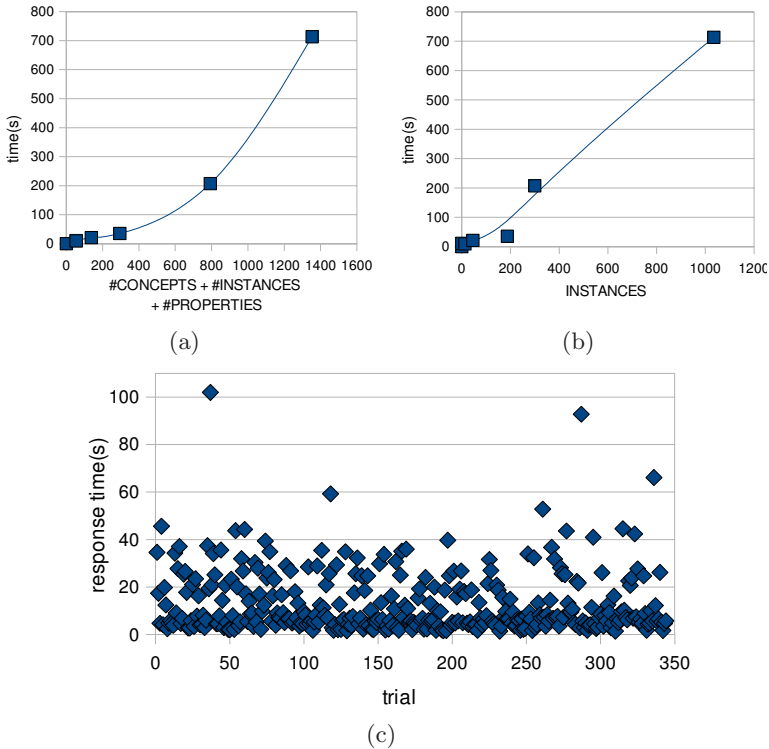


Fig. 4. Tagonto performance

6 Related Work

The related works we consider can be grouped in (i) approaches which use ontologies to describe the domain knowledge and (ii) those which use ontologies to describe the tag system itself. SOBOLEO (SOcial BOokmarking and Lightweight Engineering of Ontologies, [18]) is a tool which allows to tag resources in the Web using ontology concepts and interacting with the ontology, modifying concept labels and relations. The SOBOLEO approach shares *TagOnto* objectives, but tries to exploit directly the ontology concepts as tags. [4] suggests an integrated approach to build ontologies from folksonomies, combining statistical analysis, online lexical resources, ontology matching, and community based consensus management. [1] presents an approach to enrich the tag space with semantic relations, “harvesting the Semantic Web”, by using the tool [16]. [3] addresses the problem of translating a folksonomy into a lightweight ontology in a corporate environment by exploiting the Levenshtein metric, co-occurrence, conditional probability, transitive reduction, and visualization. [15] uses the SIOC ontology in order to represent connections between tags and concepts of a domain ontology. [11] maps tags from del.icio.us with concepts from WordNet, and uses this mapping to provide an alternative interface for browsing tags. Gruber [9,8] models the act of tagging as a quadruple (resource, tag, user, source/context) or a quintuple with a polarity argument, allowing to bind tagging data according to one particular system. Thus, tags from different systems can coexist in this model and it is possible to specify relations between them, allowing better interoperability. [14] defines a tag ontology to describe the tagging activity and the relationships between tags. [10] presents SCOT, an ontology for sharing and reusing tag data and for representing social relations among individuals. The ontology is linked to SIOC, FOAF and SKOS to link information respectively to resources, people and tags. [5] proposes a method to model folksonomies using ontologies. The model consists of an OWL ontology, capable of defining not only the main participants in the tagging activity, but also complex relations that describe tag variations (like *hasAltLabel* or *hasHiddenLabel*).

7 Conclusions

In this paper we presented *TagOnto*, a *folksonomy aggregator*, combining the collaborative nature of Web2.0 with the semantic features provided by ontologies, to improve the user experience in searching and browsing the web. The design of the system has been such that very limited overhead is imposed to users and content providers to enable these new features. *TagOnto* key components are a multi-folksonomy, tag-based search engine, and an ontology-based query refinement facility, which exploits a domain ontology to filter results and to focus users’ attention. In the best Web2.0 tradition, these features are delivered through an intuitive and reactive AJAX interface. The system is released and demonstrated online, and has been successfully tested on several domains. Nonetheless, we consider *TagOnto* a starting point for further developments and

we plan to devote more work on three key aspects: usability, performance and extensibility.

References

1. Angeletou, S., Sabou, M., Specia, L., Motta, E.: Bridging the gap between folksonomies and the semantic web: An experience report. In: Bridging the Gap between Semantic Web and Web 2.0 (SemNet 2007), pp. 30–43 (2007)
2. Curino, C., Orsi, G., Tanca, L.: X-som: A flexible ontology mapper. In: DEXA Int. Workshop on Semantic Web Architectures For Enterprises (SWAE) (2007)
3. Van Damme, C., Coenen, T., Vandijck, E.: Turning a corporate folksonomy into a lightweight corporate ontology. In: BIS. Lecture Notes in Business Information Processing, vol. 7, pp. 36–47. Springer, Heidelberg (2008)
4. Van Damme, C., Hepp, M., Siorpaes, K.: Folksontology: An integrated approach for turning folksonomies into ontologies. In: Bridging the Gap between Semantic Web and Web 2.0 (SemNet 2007), pp. 57–70 (2007)
5. Echarte, F., Astrain, J.J., Crdoba, A., Villadangos, J.E.: Ontology of folksonomy: A new modelling method. In: SAAKM, CEUR Workshop Proceedings, vol. 289 (2007)
6. Euzenat, J., Shvaiko, P.: Ontology matching. Springer, Heidelberg (DE) (2007)
7. Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures. PhD thesis, University of California, Irvine (2000)
8. Gruber, T.: Ontology of folksonomy: A mash-up of apples and oranges (2005), <http://tomgruber.org/writing/ontology-of-folksonomy.htm>
9. Gruber, T.: Tagontology - a way to agree on the semantics of tagging data (2005), <http://tomgruber.org/writing/tagontology-tagcamp-talk.pdf>
10. Kim, H.L., Breslin, J.G., Yang, S.-K., Kim, H.-G.: Social semantic cloud of tag: Semantic model for social tagging. In: Nguyen, N.T., Jo, G.S., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2008. LNCS (LNAI), vol. 4953, pp. 83–92. Springer, Heidelberg (2008)
11. Laniado, D., Eynard, D., Colombetti, M.: Using wordnet to turn a folksonomy into a hierarchy of concepts. In: Semantic Web Application and Perspectives - Fourth Italian Semantic Web Workshop, pp. 192–201 (December 2007)
12. Leacock, C., Chodorow, M., Miller, G.A.: Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics* 24(1), 147–165 (1998)
13. Liu, L., Oszu, M.T.: *Encyclopedia of Database Systems*. Springer, Heidelberg (2008)
14. Newmann, R.: Tag ontology design (2005), <http://www.holygoat.co.uk/projects/tags/>
15. Passant, A.: Using Ontologies to Strengthen Folksonomies and Enrich Information Retrieval in Weblogs. In: Proceedings of the First International Conference on Weblogs and Social Media (ICWSM), Boulder, Colorado (March 2007)
16. Sabou, M., d’Aquin, M., Motta, E.: Using the semantic web as background knowledge for ontology mapping. In: Ontology Matching, CEUR Workshop Proceedings, vol. 225 (2006)
17. Wander Wal, T.: Definition of folksonomy. In: Online blog post at (2004), <http://www.vanderwal.net/folksonomy.html>
18. Zacharias, V., Braun, S.: Soboleo – social bookmarking and lightweight engineering of ontologies. In: CKC, CEUR Workshop Proceedings, vol. 273 (2007)

AWeSOMe 2008 PC Co-chairs' Message

The Fourth International Workshop on Agents and Web Services Merging in Distributed Environments (AWeSOMe 2008) was held in conjunction with the OnTheMove Federated Conferences (OTM 2008) in Monterrey, Mexico, in November 2008. AWeSOMe is an interdisciplinary workshop focusing on research and applications combining Web services, ontologies and agents, leading to the development of an intelligent service Web.

Web services are a rapidly expanding approach to building distributed software systems across networks such as the Internet. A Web service is an operation typically addressed via a URI, declaratively described using widely accepted standards, and accessed via platform-independent XML-based messages.

Agents and multi-agent systems, on the other hand, provide an efficient way to develop large-scale systems that take advantage of distributed problem solving to provide a solution to complex problems.

Both agents and Web services can greatly benefit from their combination, since they both try to solve the same kind of problem from different perspectives and at different levels of abstraction. While agents provide the mechanisms to organize large systems and solve complex problems, Web services provide a way to find and compose services to solve new problems using existing pieces.

In addition, the increasing development of Service Oriented Architectures (SOA) and Grid Computing provides the increasing development a means to facilitate the combination of agents and Web services as a tool to create large-scale distributed systems from heterogeneous sources.

The Program Committee members did an excellent job in selecting high-quality papers. All submitted papers underwent a thorough review process with each paper having at least two reviewers providing feedback to the authors.

We would like to thank the members of the Program Committee who gave their time and energy to ensure the high quality of the technical program. We are grateful to the OTM 2008 organizers for their support and encouragement. Especially, we would like to thank the OTM 2008 Chairs, Robert Meersman and Zahir Tari.

We acknowledge the efforts of the authors of selected papers for their contributions to the new and exciting interdisciplinary area covered by the workshop. We also thank the authors of invited papers, whose excellent work has helped to keep the scientific quality of the workshop at a very high level.

November 2008

Joerg Denzinger
Pilar Herrero
Gonzalo Méndez
Rainer Unland

Teaching about Madrid: A Collaborative Agents-Based Distributed Learning Course

José Luis Bosque¹, Pilar Herrero², and Susana Mata³

¹Dpto. de Electrónica y Computadores. Universidad de Cantabria
joseluis.bosque@unican.es

²Facultad de Informática. Universidad Politécnica de Madrid, Spain
pherrero@fi.upm.es

³Escuela Superior de Informática, Universidad Rey Juan Carlos, Spain
susana.mata@urjc.es

Abstract. Interactive art courses require a huge amount of computational resources to be running on real time. These computational resources are even bigger if the course has been designed as a Virtual Environment with which students can interact. In this paper, we present an initiative that has been developed in a close collaboration between two Spanish Universities: Universidad Politécnica de Madrid and Universidad Rey Juan Carlos with the aim of joining two previous research projects: a Collaborative Awareness Model for Task-Balancing-Deliberation (CAMT) in clusters and the “Teaching about Madrid” course, which provides a cultural interactive background of the capital of Spain.

Keywords: Cluster computing, task assignment, collaborative work.

1 Introduction

The “Teaching about Madrid” course was developed with the aim of designing a virtual tour around Madrid. This course was composed by a set of interactive images (see Figure 1) that were presented in real time by a tour-guide. Students could interact with the scenario, if need, to get more specific information about the monument, such as the year in which it was built. Students can also collaborate with each other to learn together from the projected environment. Each of the scenarios is projected on a CAVE governed by cluster of PCs. A high speed myrinet network allows processing all these operations in real time.

As for the CAMT model, it was the result of a previous collaboration between the Universidad Politécnica de Madrid and the Universidad Rey Juan Carlos. This has been designed based on the extension and reinterpretation of one of the most successful models of awareness in Computer Supported Cooperative Work (CSCW), called the Spatial Model of Interaction (SMI), which manages awareness of interaction in collaborative distributed systems, through a multi-agent architecture to create a collaborative and cooperative environment. CAMT manages the interaction in the environment allowing the autonomous, efficient and independent task allocation in the environment.



Fig. 1. Real Palace of Madrid

This paper also presents how the CAMT model complements to the “Teaching about Madrid” course as it select the best processor to make the complex render task of each of the images of the course in the cluster. CAMT divide this render task of each of these images in a set of independent processes which are assigned to the more suitable nodes in the cluster. The CAMT model’s algorithms achieve very important improvements with respect to the response time and speedup.

2 Related Work

A taxonomy of load balancing methods has been defined in [3], taking into account different aspects. Three important criteria for this classification are: *Time in which workload distribution is performed* static [6] or dynamic [11]; *Control* which can be centralized [10] or distributed [6] and *System state view* global [6] or local [4].

One approach is presented in [15], which defines a generic and scalable architecture for the efficient use of resources in a cluster based on CORBA. DASH (Dynamic Agent System for Heterogeneous) [13] is an agent-based architecture for load balancing in heterogeneous clusters. The most noticeable characteristic of this proposal is the definition of a collaborative awareness model, used for providing global information that helps establish a suitable load balance. Unlike this work, our proposal (CAMT) extends and reinterprets one of the most successful models of awareness, the Spatial Model of Interaction (SMI), which manages awareness of interaction through a set of key concepts. Most of the agent-based load balancing systems use mobile agents, which makes easier the migration of tasks [7, 14].

3 Reinterpreting the SMI Key Concepts

The Spatial Model of Interaction (SMI) [2] is based on a set of key concepts which are abstract and open enough as to be reinterpreted in many other contexts with very different meanings. The model itself defines five linked concepts: medium, focus, nimbus, aura and awareness.

Medium: A prerequisite for useful communication is that two objects have a compatible medium in which both objects can communicate. *Aura*: The sub-space which effectively bounds the presence of an object within a given medium and which acts as an enabler of potential interaction [5]. In each particular medium, it is possible to delimit the observing object's interest. This idea was introduced by S. Benford in 1993, and it was called *Focus*. In the same way, it is possible to represent the observed object's projection in a particular medium, called *Nimbus*. Finally, *Awareness* quantifies the degree, nature or quality of interaction between two objects. Awareness between objects in a given medium is manipulated via Focus and Nimbus, requiring a negotiation process.

Let's consider a system containing a set of nodes $\{n_i\}$ and a task T that requires a set of processes to be solved in the system. Each of these processes necessitates some specific requirements being r_i the set of requirements associated to the process p_i , and therefore each of the processes will be identified by the tuple (p_i, r_i) . The CAMT model reinterprets the SMI key concepts as follow:

Focus: It is interpreted as the subset of the space on which the user has focused his attention with the aim of interacting with.

Nimbus: It is a tuple ($Nimbus = (NimbusState, NimbusSpace)$) containing information about: (a) the load of the system in a given time (*NimbusState*); (b) the subset of the space in which a given node projects its presence (*NimbusSpace*). As for the *NimbusState*, this concept will depend on the processor characteristics as well as on the load of the system in a given time. In this way, the *NimbusState* could have three possible values: *Null*, *Medium* or *Maximum*.

Awareness of Interaction (AwareInt): This concept will quantify the degree, nature or quality of asynchronous interaction between distributed resources. Following the awareness classification introduced by Greenhalgh in [8], this awareness could be *Full*, *Peripheral* or *Null*.

$$AwareInt(n_i, n_j) = Full \quad \text{if } n_j \in Focus(\{n_i\}) \quad \wedge \quad n_i \in Nimbus(n_j)$$

Peripheral aware of interaction if

$$AwareInt(n_i, n_j) = Peripheral \quad \text{if } \begin{array}{l} n_j \in Focus(\{n_i\}) \quad \wedge \quad n_i \notin Nimbus(n_j) \\ \text{or} \\ n_j \notin Focus(\{n_i\}) \quad \wedge \quad n_i \in Nimbus(n_j) \end{array}$$

The CAMT model is more than a reinterpretation of the SMI, it extends the SMI to introduce some new concepts such us:

Interactive Pool: This function returns the set of nodes $\{n_j\}$ interacting with the n_i node in a given moment.

Task Resolution: This function determines if there is a service in the node n_i , being $NimbusState(n_i) \neq Null$, such that could be useful to execute the task T (or at least one

of its processes). This concept would also complement the Nimbus concept, because the NimbusSpace will determine those machines that could be taking into account in the tasks assignment process because they are not overload yet.

Collaborative Organization: This function will take into account the set of nodes determined by the *InteractivePool* and will return those nodes of the System in which it is more suitable to execute the task T (or at least one of its processes p_i). This selection will be made by means of the *TaskResolution* function.

4 The Load Balancing Algorithm in CAMT

The main characteristics of this algorithm are that it is dynamic, distributed, global and take into account the system heterogeneity. This algorithm contents the following policies [12].

State Measurement Rule: It is in charge of getting information about the computational capabilities of the node in the system. This information, quantified by a load index, provides aware of the NimbusState of the node. Several authors have studied their effects on the system performance [9]. However, as for the CPU utilization, we are especially interested on the computational capabilities of the node for the new task to be executed. In this research work the concept of CPU assignment is use to determine the load index. The CPU assignment, is defined as the CPU percentage that can be assigned to a new task to be executed in the node N_i . The calculation of this assignment is based on two dynamic parameters: the number of tasks N, which are ready to be executed in the CPU queue and the percentage of occupation of the CPU, U, and it would be calculated as:

$$\text{If } (U \geq \frac{1}{N}) \Rightarrow A_{CPU} = \frac{1}{N+1}$$

$$\text{If } (U < \frac{1}{N}) \Rightarrow A_{CPU} = 1 - U$$

The NimbusState of the node depends on the load index value and an increase or decrease of this index over a specific threshold will imply the corresponding modification in the NimbusSate. It determines if the node could execute more, local or remote, tasks. Its possible values would be:

- *Maximum:* The load index is low, this node will execute all the local tasks, accepting all new remote execution requests coming from other nodes.
- *Medium:* The load index has an intermediate value and therefore the node will execute all the local tasks, but it cannot execute remote tasks.
- *Null:* The load index has a high value and therefore the node is overload. In this situation, the node will not execute new tasks.

Information exchange rule: The knowledge of the global state of the system will be determined by a policy on the information exchange. This policy should keep the information coherence without overloading the network with an excessive number of unnecessary messages. An optimum information exchange rule for the CAMT model should be based on events [1]. This rule only collects information when a change in the Nimbus of the nodes is made. If later, the node that has modified its nimbus will

be in charge of notifying this modification to the rest of the nodes in the system, avoiding thus synchronisation points. As this algorithm is global, this information has to be sent to all the nodes in the system.

Initiation rule: As the model implements a non user interruption algorithm, the selection of the node must be made just before sending the task execution. The decision of starting a new load balancing operation is completely local, depends on the local information storage. when a node intends to throw the execution of a new task, the initialization rule will evaluate:

If (NimbusState = Null), a new load balancing operation is started.

Load Balancing Operation: Once the node has made the decision of starting a new load balancing operation, this operation will be divided in another three different rules: localization, distribution and selection.

The localization rule has to determine which nodes are involved in the CollaborativeOrganization of the node n_i . In order to make it possible, firstly, the CAMT model will need to determine the awareness of interaction of this node with those nodes inside its focus. Those nodes whose awareness of interaction with n_i was Full will be part of the InteractivePool of n_i to solve the task T, and from that pre-selection the TaskResolution method will determine those nodes that are suitable to solve efficiently the task in the environment.

This algorithm joins selection and distribution rules because it determines which nodes (among all the nodes constituting the CollaborativeOrganization) will be in charge of executing each of the processes making up the T task. The goal of this algorithm is to find the more equilibrate processes assignment to the computational nodes, based on a set of heuristics. Firstly, a complete distribution of the processes making up the T task is made in the computational nodes implicated in the CollaborativeOrganization. If, in this first turn, all the process would be assigned to one of the nodes involved in the CollaborativeOrganization, the algorithm would have finished.

5 The Underlying Architecture

The load balancing multi-agent architecture is composed of four agents which are replicated for each of the nodes of the cluster.

Load Agent: The Load Agent (LA) calculates, periodically, the load index of the local node and evaluates the changes on its state. Moreover, it defines the thresholds determining the changes on its state for that node. When it detects a change on the state, this modification is notified to the local GSA and IA. The first step of the LA is to obtain the static information. Then this information is communicated to the rest of the nodes through the MPI_Reduce function, which is in charge of calculating the maximum of the computational power of all the nodes composing the cluster.

Next, the agent enters in an infinite loop where it gets dynamically information and calculates the new state. With the new state the agent determines if a node state change has occurred and communicates it to the local GSA and IA.

Global State Agent (GSA): This agent implements the exchange information rule, and therefore its main functionality is to manage the state information exchanged

among the nodes of the system, and provide LBA with this information. Firstly, it determines the current InteractivePool. Next, the agent enters in an infinite loop in which it is waiting for receiving messages from other agents. These messages are:

- **LOCAL_STATE_CHANGE:** This message comes from the local LA and this information has to be notified to all the Global State Agents that are located in a different node of the cluster to update their lists.
- **REMOTE_STATE_CHANGE:** In this case, only the local state list should be modified to update the new state of the remote node.
- **INTERACTIVE_POOL_REQUEST:** The local LBA request the Interactive-Pool to the GSA. The GSA responds to this request providing it with the required information.

Initiation Agent (IA): This agent is in charge of evaluating the initialisation rule and it determines, if the task can be executed locally or if a new load balancing operation has to be carried out. Its main structure contains an infinite loop and, for each of these iterations, the pending tasks in the execution queue are checked. If there is a pending task, a new assignment task loop starts:

- **LOCAL_STATE_CHANGE:** It receives a message from the local LA to notify a change on the local state.
- **EXECUTE_TASK_REQUEST:** It requests execution of a new task. As a task is composed by a set of processes, the local execution of one of these processes can change the NimbusState of that node. Therefore, when an execution request is received, the IA starts a loop to assign all the processes of the task. For the first process, the NimbusState is checked to corroborate if its value is equal to Maximum. If later, that process is executed locally. On the other hand, a new balancing operation would start and a message would be sent to the local LBA.

Load Balancer Agent (LBA): This agent is responsible of making the load balancing operation. Its structure contains an infinite loop that is waiting to receive messages from other agents, being the possible messages:

- **BALANCER EXECUTION:** This message comes from the local IA and it indicates that a new load balancing operation needs to start. For the localization rule, the LBA will follow the following sequence of steps:
 1. Request the InteractivePool and the states list to the local GSA.
 2. Determine the TaskResolution, analysing which nodes of the InteractivePool have their NimbusState different to Null.
 3. Request the score, of those processes composing the task to be executed, to the LBA of the nodes included in the TaskResolution
 4. Taking into account the TaskResolution and the requested scores, determine the Collaborative_Organization by analysing those nodes that, belonging to the TaskResolution, can execute at least one of the processes of the task.

As for the selection and distribution rule, once the CollaborativeOrganization has been made up, it is necessary to determine which processes are sent to each of the nodes of the cluster. In order to make this possible, the algorithm presented in

section 4 has been implemented. Once all the process had been assigned, they would be sent to the designated nodes.

- **REMOTE_EXECUTION:** The message received comes from the remote LBA, asking for the remote execution of a process. Once the LBA has checked its own state, it replies to the remote LBA with an acceptance or rejection message. If the process is accepted, the operation would conclude, the LBA would execute the process locally and it would update its state. The rejection could be due to a change on its nimbusState (to Null) which has not been notified yet due to the network latency.

6 AMT Evaluation on the “Teaching about Madrid Course”

The “Teaching about Madrid” course requires the render of realistic scenarios, in real time, for an immersive environment. This task entails complex processes - such as geometric transformations, collision detection, and illumination and shadowing algorithms- that require a huge amount of floating point mathematical operations. On the other hand, the CAVE has 4 projectors and each of these projectors are connected to a different PC which is in charge of rendering the corresponding images of the scenario. However, as the geometrical model and the illumination algorithms are getting more complex, the computational capacity of these PCs gets overflowed and some images are lost. If later, users perceive a gap between two consecutive images and therefore the scenario’s realism and the user’s immersion decreases considerable.

On the other hand, as the render task can be split up in several processes which can be executed independently, the CAMT model seems to be appropriated for improving the “Teaching about Madrid” performance through the execution of the render task in a high-performance cluster.

The cluster is made up of 40 PCs (nodes) connected through a 1.1 Gbps Myrinet Network. Each cluster node is a 2 GHz AMD K7 processor with 512 MB of main



Fig. 2. Puerta del Sol of Madrid



Fig. 3. Teatro Real

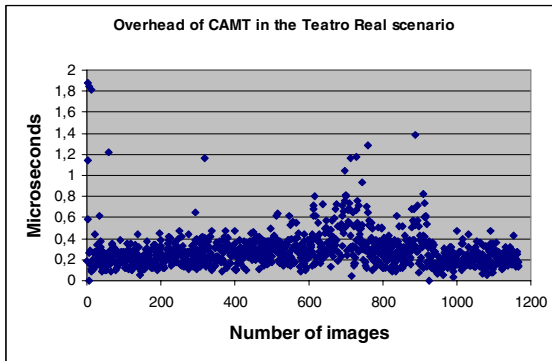


Fig. 4. CAMT Overhead in the Teatro Real

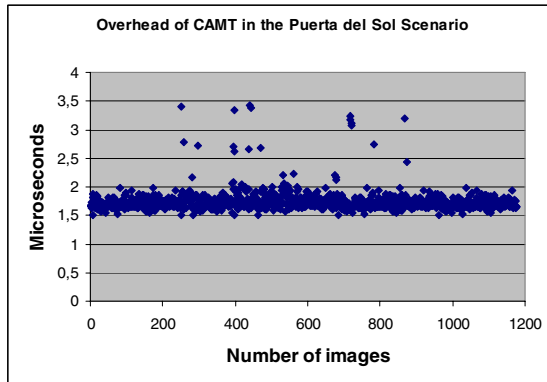


Fig. 5. CAMT Overhead in the Puerta del Sol

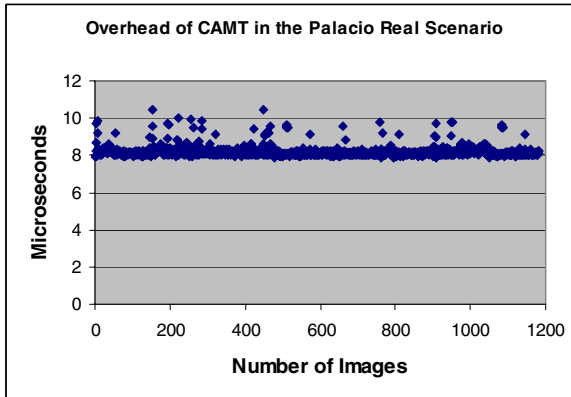


Fig. 6. CAMT Overhead in the Palacio Real

memory. The CAMT model has been developed using GNU tools and LAM/MPI 7.1.1 Library. As a way of evaluating the improvement that the CAMT model introduces on the performance of the Teaching about Madrid course, we have selected few scenarios with different level of geometrical complexity. These scenarios are: Teatro Real (Figure 3), Puerta del Sol (Figure 2) and Palacio Real (Figure 6).

In this section we present the set of results obtained evaluating the overhead introduced by CAMT while it assign the processes to the nodes of the cluster. As for the course overhead, Figures 4, 5 and 6 demonstrate that the overhead incurred by the algorithm to assign a process doesn't interfere with the frame's rate of the CAVE's projectors. The overhead remains almost constant for all of the tasks and processes even although it increases as the geometrical complexity of the scenario - and therefore the data file's size- also increases, demonstrating that the CAMT algorithm has been endowed with very strong scalability features.

7 Conclusions

This paper presents the integration of two previous research works. The first of these two projects was a guided course, named "Teaching about Madrid" which intended to provide students with a cultural interactive background of Madrid. The second one, CAMT, manages awareness of interaction in collaborative distributed systems, through a multi-agent architecture to allow the autonomous, efficient and independent task allocation in the environment. The CAMT model complements to the "Teaching about Madrid" course as it select the best processor to make the complex render task of each of the images of the course in the cluster. CAMT divide this render task of each of these images in a set of independent processes which are assigned to the more suitable nodes in the cluster. Thus, even although the geometrical model and the illumination algorithms are complex, practically none of the images are lost, and users never perceive a gap between two consecutives images, feeling a high degree of realism and immersion. Moreover, the experimental results presented in this paper demonstrate that the overhead incurred by the algorithm to assign a process doesn't

interfere with the frame's rate of the CAVE's projectors, and therefore we can conclude that CAMT complements successfully the teaching course.

References

- [1] Beltrán, M., Bosque, J.L., Guzmán, A.: Resource Dissemination policies on Grids. In: *On the Move to Meaningful Internet Systems 2004: OTM 2004*. LNCS, pp. 135–144. Springer, Heidelberg (2004)
- [2] Benford, S.D., Fahlén, L.E.: A Spatial Model of Interaction in Large Virtual Environments. In: *Proceedings of the Third European Conference on Computer Supported Cooperative Work (ECSCW 1993)*, Milano, Italy, pp. 109–124 (1993)
- [3] Casavant, T.L., Kuhl, J.G.: A taxonomy of scheduling in general-purpose distributed computing systems. In: *Readings and Distributed Computing Systems*, pp. 31–51 (1994)
- [4] Corradi, A., Leonardi, L., Zambonelli, F.: Diffusive load-balancing policies for dynamic applications. *IEEE Concurrency* 7(1), 22–31 (1999)
- [5] Darbha, S., Agrawal, D.P.: Optimal scheduling algorithm for distributed-memory machines. *IEEE Trans. on Parallel and Distributed Systems* 9(1), 87–95 (1998)
- [6] Das, S.K., Harvey, D.J., Biswas, R.: Parallel processing of adaptive meshes with load balancing. *IEEE Trans. on Parallel and Distributed Systems* (12), 1269–1280 (2001)
- [7] Desic, S., Huljenic, D.: Agents based load balancing with component distribution capability. In: *Proceedings of the 2nd International Symposium on Cluster Computing and the Grid (CCGRID 2002)* (2002)
- [8] Greenhalgh, C.: *Large Scale Collaborative Virtual Environments*, Doctoral Thesis. University of Nottingham (October 1997)
- [9] Kunz, T.: The influence of different workload descriptions on a heuristic load balancing scheme. *IEEE Transactions on Software Engineering* 17(7), 725–730 (1991)
- [10] Schnor, B., Petri, S., Oleyniczak, R., Langendorfer, H.: Scheduling of Parallel Applications on Heterogeneous Workstation Clusters. In: *Proc. of the 9th Int. Conf. on Parallel and Distributed Computing Systems*, vol. 1, pp. 330–337 (September 1996)
- [11] Xiao, L., Chen, S., Zhang, X.: Dynamic cluster resource allocations for jobs with known and unknown memory demands. *IEEE Trans. on Parallel and Distributed Systems* 13(3), 223–240 (2002)
- [12] Xu, C., Lau, F.: *Load balancing in parallel computers: theory and practice*. Kluwer Academic Publishers, Dordrecht (1997)
- [13] Rajagopalan, A., Hariri, S.: An Agent Based Dynamic Load Balancing System. In: *Proc. of the Int.l. Workshop on Autonomous Decentralized Systems*, pp. 164–171 (2000)
- [14] Suri, N., Groth, P.T., Bradshaw, J.M.: While You're Away: A System for Load-Balancing and Resource Sharing Based on Mobile Agents. In: *1st Int. Sym. on Cluster Computing and the Grid*, p. 470 (2001)
- [15] Vanhastel, S., De Turck, F., Demeester, P.: Design of a generic platform for efficient and scalable cluster computing based on middleware technology. In: *Proceedings of the CCGRID 2001*, pp. 40–47 (2001)

The Principle of Immanence in GRID-Multiagent Integrated Systems

Pascal Dugenie¹, Clement Jonquet^{1,2}, and Stefano A. Cerri¹

¹ Laboratory of Informatics, Robotics, and Microelectronics of Montpellier (LIRMM)
National Center of Scientific Research (CNRS) & University Montpellier 2
161 Rue Ada, 34392 Montpellier, France

{dugenie, cerri}@lirmm.fr

² Stanford Center for Biomedical Informatics Research (BMIR)
Stanford University School of Medicine
Medical School Office Building, Room X-215
251 Campus Drive, Stanford, CA 94305-5479 USA
jonquet@stanford.edu

Abstract. Immanence reflects the principle of emergence of something new from inside a complex system (by opposition to transcendence). For example, immanence occurs when social organization emerges from the internal behaviour of a complex system. In this position paper, we defend the vision that the integration of the GRID and Multi-Agent System (MAS) models enables immanence to occur in the corresponding integrated systems and allows self-organization. On one hand, GRID is known to be an extraordinary infrastructure for coordinating distributed computing resources and Virtual Organizations (VOs). On the other hand MAS interest focusses on complex behaviour of systems of agents. Although several existing VO models specify how to manage resource, services, security policies and communities of users, none of them has considered to tackle the internal self-organization aspect of the overall complex system. We briefly present AGORA, a virtual organization model integrated in an experimental collaborative environment platform. AGORA's architecture adopts a novel design approach, modelled as a dynamic system in which the result of agent interactions are fed back into the system structure.

1 Introduction

Immanence usually refers to philosophical and metaphysical concepts. However, immanence expresses, in a wider sense, the idea of a strong interdependence between the *organization* and the *activity* of a complex system. An system is immanent if it constantly re-constructs its own structural organization throughout its internal activity: *The organisation is immanent to the activity*. By opposition, a system whose behaviour would be completely determined from the initial conditions with no feedback effect of its activity on its own structure is not an immanent system and has no chance to be self adaptive in case of changes of conditions of its environment. Examples of immanent systems are living systems in biology or social organizations. The principle of immanence has been introduced in informatics to describe the social impact derived from

the introduction of the internet in the society. Furthermore, the expression *collective intelligence* has been adopted to describe the immanent system of knowledge structured around the Web [1]: The material (*i.e.*, the Web) is immanent to the immaterial (*i.e.*, the collective intelligence).

The notion of immanence, which was appearing quite utopic only a few years ago, is gaining an increasing interest in informatics because of technological maturity to demonstrate its feasibility. Immanence is becoming highly critical in the analysis of complex problems and therefore, there are several reasons for considering the potential of immanence in collaborative environments based on the integration of GRID and Multi-Agent System (MAS):

One reason is the possibility offered by the GRID infrastructure to deploy autonomous services [2].¹ These services can be instantiated in specific service container with their dedicated resources, and adopt a proactive behaviour. This is a major difference between the GRID over the Web which is not able to provide stateful resources necessary to operate autonomous services.

Another reason is the trend for the holistic approach for modelling collective behaviour in MAS. In this approach, interactions between agents are contextualized within a global collaborative process. The notion of agent is extended to cover artificial processes as well as human ones. Agents interact within a collaborative environment by providing or using services. Moreover, they can behave intelligently with those services. One essential condition for a collaborative environment to become immanent is that any agent of the system may play an active role in the system construction [3]. For instance, both system designers and expert-users have a symmetrical feedback in the cycle of developing and validating a complex application. They interact by providing services to each other via a common collaboration kernel. They may develop their point of view in the context of a collaboration process and their role may evolve indefinitely. Thus, such a system clearly requires self-adaptiveness and self-organization.

In this paper, we defend the vision that the integration of the GRID and Multi-Agent System (MAS) models enables immanence to occur in the corresponding integrated systems and allows self-organization. In such systems, immanence constitutes the *living link* between the organization (*i.e.*, the static model) and the activity (*i.e.*, the dynamic model), and both models act upon each other. The organization enables to generate the activity whereas the activity constantly seeks to improve the organization. To illustrate our vision, we briefly present the AGORA *ubiquitous collaborative space*, a virtual collaborative environment platform that benefits from an original immanent Virtual Organization (VO) management system thanks to its GRID-MAS based underlying model.

2 A Brief State-of-the Art

Most of the GRID research activity eludes the question of immanence because two complementary aspects are usually treated separately while they should be treated together. The *collaboration* aspect is treated in general within the domain of Computer

¹ An service-oriented architecture has been adopted in the Open Grid Service Architecture that has become the reference model for GRID systems.

Supported Collaborative/Cooperative Work. The *VO management* aspect is studied through various conceptual models of organization.

2.1 Virtual Collaborative Environments

Virtual Collaborative Environments (VCEs) are mainly studied in the domain of Computer Supported Collaborative/Cooperative Work. A VCE is a set of distributed computing resources, services and interfaces aiming to provide an advanced environment for collaboration. These interfaces may support several modalities of communication such as audio-video, shared visualization, instant messaging, notification and shared file repositories. The multiplicity of modalities of communication (*i.e.*, multimodal communication) lays on the principle that increasing the *awareness* improves the efficiency of the collaboration process. The advantage of a GRID infrastructure for a VCE is to allow a seamless access to distributed computing resources. Furthermore, this access benefits from the principle of *ubiquity* since GRID resources are able to maintain the state of the communications independently from the location of the terminal elements.

Access Grid (www.accessgrid.org) is the most world-wide deployed GRID VCE. Access Grid operates on *Globus* [4], the most popular GRID middleware. The topology of the Access Grid infrastructure consists of two kinds of nodes [5]: the venue clients and the venue servers. Access Grid venue clients can meet in a venue server to set up a meeting. Access Grid uses the H.263 protocol [6] for audio and video encoding and multicast method to distribute the communication flow between sites. The display of multiple H.263 cameras in every site gives a strong feeling of presence from every other site. The modular characteristic of Access Grid allows to add new features such as application sharing (shared desktop, presentation, etc.) and data sharing. Access Grid focusses on the principles of *awareness* and *ubiquity*. However, Access Grid does not include a powerful mean for VO management. VO are managed in an *ad hoc* manner at the venue server side. This has the inconvenience to require much technical administrative work from computer experts in this domain. Therefore, Access Grid has no potential for immanence.

2.2 Virtual Organization Management Models

In its original definition, a VO is a community of users and a collection of virtual resources that form a coherent entity with its own policies of management and security [7]. A rudimentary VO management system has been originally built-in *Globus* but has little potential for scalability. In order to resolve these limitations several VO management models have been proposed within the GRID community. For examples:

Community Authorization Service (CAS) has been specifically designed to facilitate the management of large VO [8]. The functionalities for VO membership and rights management are centralized in a Lightweight Directory Access Protocol (LDAP) directory. Since, the structure of the VO is strongly hierarchical, this is hardly possible to reorganize the initial tree once the services are deployed.

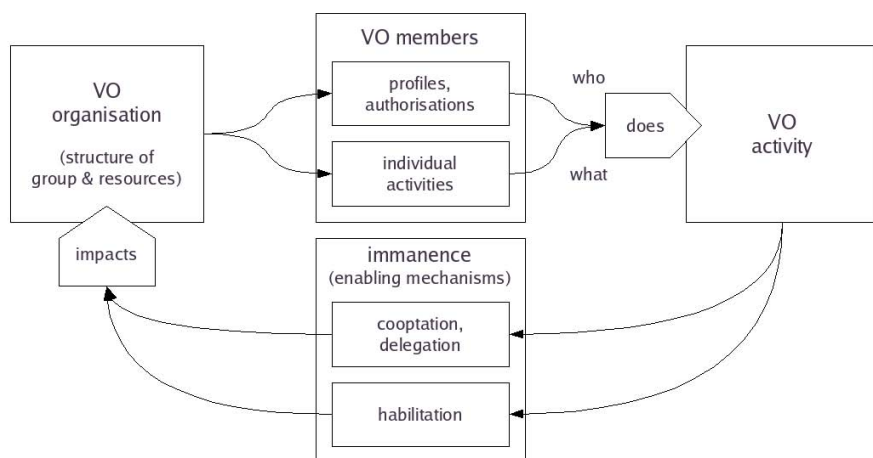


Fig. 1. A typical VO management model

Virtual Organization Membership Service (VOMS) [9] is deployed in more recent GRID infrastructures such as in the EGEE project (<http://eu-egee.org/>). It resolves some problems of CAS such as the membership management by providing a more evolutive relational database instead of a flat tree structure. However, VOMS still presents some conceptual limitations such as an inheritance link between a parent VO and its children. The subdivision of VO into groups often creates confusion in the management of rights and does not enable a complete independence between the groups and the VO. For instance, the lifetime of a group is determined by the lifetime of the parent VO.

Designing an architecture allowing access to the resources of a VCE is a real technological challenge. Indeed, the models presented here are based on client-server architecture with several points of rigidity. It results that the end-user may face usability constraints related to technological choices adopted more or less arbitrarily by the designer of the architecture.

Figure 1 represents a typical VO management system. It illustrates why both CAS and VOMS fail in ensuring self-organization of VOs. On the left part, the VO management system determines the overall system organization. On the middle part, mechanisms for VO member management and, on the right part, the resulting activity is directly dependent on the initial organization. At this stage, there is no more possibility to re-introduce activities back to the system organization. The bottom part of the figure represents the mechanisms for enabling the principle of immanence. It includes processes such as *cooptation* (a set of protocols to introduce new members), *right delegation* between VO members and *habilitation* to perform tasks in the context of the VO. This involves many kinds of knowledge transfer mechanisms that are ensured during the collaboration activity.

2.3 GRID and MAS Convergence

The GRID and MAS communities believe in the potential of GRID and MAS to enhance each other because these models have developed significant complementarities [10].

One of the crucial explorations concerns the substitution by an agent-oriented kernel of the current object-oriented kernel of services available in service-oriented architectures. The community agrees that such a change will really leverage service scenarios by providing new types of services [11]. This key concept of *service* is clearly at the intersection of the GRID and MAS domains and motivate their integration [12]. In [13], we introduce the *Agent-Grid Integration Language* (AGIL) as a GRID-MAS integrated systems description language which rigorously formalizes both key GRID and MAS concepts, their relations and the rules of their integration with graphical representations and a set-theory formalization. AGIL concepts are used in section 3 to illustrate the Agora ubiquitous collaborative space architecture (figure 3). AGIL represents an integration model in which we consider agents exchanging services through VOs they are members of: both the service user and the service provider are considered to be agents. They may decide to make available one of their capabilities in a certain VO but not in another. The VO's service container is then used as a service publication/retrieval platform. A service is executed by an agent with resources allocated by the service container. We sum-up here AGIL's two main underlying ideas:

- The representation of agent capabilities as Grid services in service containers, i.e., viewing Grid service as an 'interface' of an agent capability;
- The assimilation of the service instantiation mechanism – fundamental in GRID as it allows Grid services to be stateful and dynamic – with the mechanism to create dedicated conversation contexts fundamental in MASs.

Instead of being centred on how each entities of the system should behave, both GRID and MAS have chosen an organizational perspective in their descriptions. In organization centred MAS [14][15], the concepts of organizations, groups, roles, or communities play an important role. In particular, Ferber et al. [15] presents the main drawbacks of agent-centred MAS and proposes a very concise and minimal organization centred model called Agent-Group-Role (AGR) from which AGIL is inspired as summarized in table 1. GRID-MAS integrated system benefit from both GRID and MAS organizational structure formalisms. Therefore, the convergence of GRID and MAS research activities brings up new perspectives towards a immanent system.

3 The AGORA Ubiquitous Collaborative Space

AGORA is an original VO model which exhibits the principles of *immanence*, *ubiquity* and *awareness*. Moreover, for experimental purposes, the VCE platform called AGORA Ubiquitous Collaborative Space (UCS) has been implemented and deployed four years ago in the context of the European project ELeGI² when the participants could not identify a VCE on GRID that minimize the number of intervention of software specialists. In order to demonstrate the effectiveness of the solution, extensive experiments of the AGORA UCS prototype have been performed with more than eighty users across the world [16] (cf. section 3.3).

² ELeGI (European Learning Grid Infrastructure) project, 2004-2007, www.elegi.org

Table 1. AGIL's organizational-structure analogies between GRID and MAS

MAS	GRID
Agent	Grid user
An agent is an active, communicating entity playing roles and delegating tasks within groups. An agent may be a member of several groups, and may hold multiple roles (in different groups).	A Grid user is an active, communicating entity providing and using services within a VO. A Grid user may be a member of multiple VOs, and may provide or use several services (in different VOs).
Group	VO
A group is a set of (one or several) agents sharing some common characteristics and/or goals. A group is used as a context for a pattern of activities and for partitioning organizations. Two agents may communicate only if they are members of the same group. An agent transforms some of its capabilities into roles (abstract representation of functional positions) when it integrates into a group.	A VO is a set of (one or several) Grid users sharing some common objectives. A VO and the associated service container is used as a context for executing services and for partitioning the entire community of Grid users. Two Grid users may exchange (provide/use) services only if they are members of the same VO. A Grid user publishes some of its capabilities into services when it integrates into a VO.
Role	Service
The role is the abstract representation of a functional position of an agent in a group. A role is defined within a group structure. An agent may play several roles in several groups. Roles are local to groups, and a role must be requested by an agent. A role may be played by several agents.	The service is the abstract representation of a functional position of a Grid user in a VO. A service is accessible via the CAS service. A Grid user may provide or use several services in several VOs. Services are local to VOs (situated in the associate container), and a Grid user must be allowed to provide or use services in a VO. A service may be provided by several Grid users.

AGORA's ubiquity principle enables access to the VCE from anywhere at anytime. Although this principle has been envisaged several years ago [17], the concrete deployment of operational solutions has been feasible only recently, by means of pervasive technologies such as GRID. AGORA's awareness principle enable enhancing presence ([18]) by means of multimodal communication such as asynchronous file repositories or messaging as well as synchronous services such as audio-video communication, text chat and shared desktop. An extensive description of these two first characteristics is out of the scope of this article. We rather focus here on the self-organization capability of the VO that is described by the third characteristic of *immanence*, which is the characteristic which really distinguishes the AGORA UCS from related works such as Access Grid.

3.1 Conceptual Model

The conceptual model of AGORA UCS presented on figure 2 consists of a set of five concepts and four relations:

Agent. This concept constitutes the main driving element that can change the state of the overall system. It's type may be a artificial (a process) or a human.

Group. This concept contains a number of agents who are considered as members of this group. An agent may be a member of several groups and plays a different activity according to the group. A group can be seen as the context of a that activity.

Organization. This concept is formed with one given group and one given set of resource. It is a bijective so that a given group is associated with a resource and one, and vice versa.

Resource. This concept is a set of means to carry out tasks. In the case of a distributed computing infrastructure such as GRID, this concept corresponds to a *container services*.³

Activity. This concept describes the way services are exchanged in the context of a group. It involves the notions of role, rights and interaction between agents.

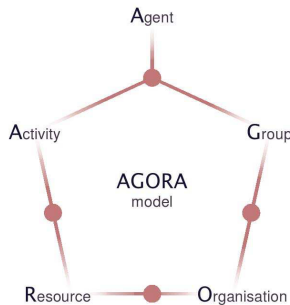


Fig. 2. AGORA UCS's conceptual model

A ternary relation between the three concepts, *agent*, *group* and *activity*, enables to resolve the limitation for self-organization of existing VO management models. This relation expresses that an agent may become member of one or several groups and play different activities in the context of one of these groups. Another important aspect of this model are the two bijective relations: one between a given *organization* and a group of agents (*i.e* a *community*) and one between this organization and a given set of resource (*i.e* a *service container*). A service container ensure the provision of resource to the *community*.

3.2 Persistent Core Services

AGORA UCS model includes a number of six persistent core services necessary for bootstrapping and maintaining a collaborative environment. Since one service container is associated to a VO, there are as many sets of persistent core services as VO. Figure 3 is a representation that uses the AGIL [13] to shows an AGORA service container including the six persistent core services:

³ GRID service containers can be defined as the reification of a portion of computing resources that have been previously virtualized through a GRID middleware.

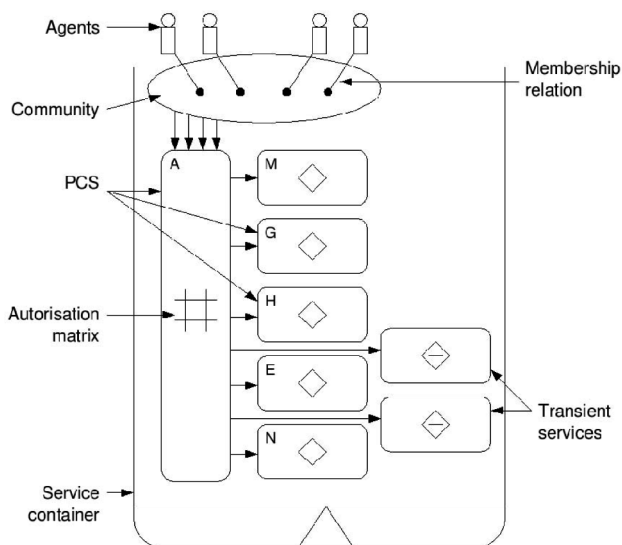


Fig. 3. AGORA UCS's persistent core services

1. **[A] uthorisations:** Members of a VO may have a different level of permission on services. This service is in charge of assigning rights to members including the permissions over the persistent core services.
2. **[M] embers:** A VO is composed of members. This service manages the description of members of a VO, adding or removing members.
3. **[G] roup:** A VO is characterized by its properties (identifier, description, etc.). Also, the creation of a new VO is always performed in the context of another VO. Therefore this service is in charge of both intra VO operations as well as extra VO operations.
4. **[H] istory:** All the data belonging to a VO must be stored, maintained and also indexed. This service is in charge of keeping track of changes, logs of events and also of recording collaboration sessions.
5. **[E] nvironment:** A VO may personalize its own environment. This environment operates in a service container. This service is in charge of adding or removing services (excluding the persistent core services).
6. **[N] otifications:** Communication between members of a VO and services is performed via notifications. This service treats the flow of notifications and manages the states of the exchanged messages.

3.3 Experimentations

Extensive experiments have allowed to validate the mechanisms for immanence by focusing on user self-ability to feel at ease in AGORA UCS. Only a simple web browser acting as a *thin terminal* is necessary. The users often noted as important the ubiquitous

access to the collaborative environment with no resource provided from their part. This allowed an immediate bootstrap of new VO and the acceptance of the technology was extremely high. The strong level of awareness allowed by the shared visualization enabled a fast transfer of knowledge in particular for mastering complex computational tools.

For instance a scenario called EnCORÉ⁴ has provided the most relevant results. AGORA UCS enabled the visual representation of chemistry models at a distance. Most attention was put by the users on the semantics of their domain rather than solving computing problems. Unskilled users, chemistry scientists, were at ease in their operations. The delegation of rights was important in the absence of some members. The cooptation of new members was also necessary to build a trustful community.

Since the behavior of a VO can not be foreseen in advance, the flexibility of the AGORA UCS is essential to enable a community to freely organize itself. Various situations of collaboration with reinforced modalities of interaction by using a synchronous communication interface have favored the transfer of knowledge. Discussions in real time, combined with visual representations on a shared desktop, allowed the actors to increase the effectiveness of the collaboration process.

4 Conclusion

Can we say that the principle of immanence has been achieved? We could answer that it partially occurred in anecdotal situations during our experiments. But the most important is that the strength of the AGORA UCS conceptual model, the experiments on the platform and all the promises of the GRID infrastructure open many significant perspectives. This work, at a very early stage, has already contributed to new ways to approach complex system design where the self-organization criteria is critical.

We are aware that a serious validation process is still necessary in order to demonstrate that the *organization* and the *activity* are completely interleaved and fully constructed each other. The convergence of GRID and Multi-Agents systems had revealed some interesting features to accomplish this view.

References

1. Lévy, P.: *Collective Intelligence: Mankind's Emerging World in Cyberspace*. Basic Books (1999)
2. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: *The physiology of the Grid: an Open Grid Services Architecture for distributed systems integration*. In: *Open Grid Service Infrastructure WG, Global Grid Forum, The Globus Alliance* (June 2002)
3. Dugénie, P., Cerri, S.A., Lemoisson, P., Gouaich, A.: *Agora UCS, Ubiquitous Collaborative Space*. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) *ITS 2008*. LNCS, vol. 5091, pp. 696–698. Springer, Heidelberg (2008)
4. Foster, I.: *Globus Toolkit Version 4: Software for service-oriented systems*. *Journal of Computer Science and Technology*, 513–520 (2006)

⁴ EnCORÉ: Encyclopédie de Chimie Organique Electronique.
Demonstration available at <http://agora.lirmm.fr>

5. Sievers, C.: Access Grid Venue Client 3.x User Manual. Technical report, Los Alamos National Laboratory (2007)
6. ITU-T: H.263, infrastructure of audiovisual services, video coding for low bit rate communication. Technical report, International Telecommunication Union (2005)
7. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the Grid: enabling scalable virtual organizations. *Supercomputer Applications* 15(3), 200–222 (2001)
8. Pearlman, L., Welch, V., Foster, I., Kesselman, C., Tuecke, S.: A Community Authorization Service for group collaboration. In: 3rd International Workshop on Policies for Distributed Systems and Networks, POLICY 2002, Monterey, CA, USA, pp. 50–59. IEEE Computer Society, Los Alamitos (2002)
9. Alfieri, R., Cecchini, R., Ciaschini, V., Dell Agnello, L., Frohner, A., Gianoli, A., Lörentey, K., Spataro, F.: An authorization system for virtual organizations. In: Fernández Rivera, F., Bubak, M., Gómez Tato, A., Doallo, R. (eds.) *Across Grids 2003*. LNCS, vol. 2970, pp. 33–40. Springer, Heidelberg (2004)
10. Foster, I., Jennings, N.R., Kesselman, C.: Brain meets brawn: why Grid and agents need each other. In: 3rd International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2004, New York, NY, USA, vol. 1, pp. 8–15 (2004)
11. Huhns, M.N., Singh, M.P., Burstein, M., Decker, K., Durfee, E., Finin, T., Gasser, L., Goriadina, H., Jennings, N.R., Lakkaraju, K., Nakashima, H., Parunak, V., Rosenschein, J.S., Ruvinsky, A., Sukthankar, G., Swarup, S., Sycara, K., Tambe, M., Wagner, T., Zavala, L.: Research directions for service-oriented multiagent systems. *Internet Computing* 9(6), 65–70 (2005)
12. Jonquet, C., Dugenie, P., Cerri, S.A.: Service-Based Integration of Grid and Multi-Agent Systems Models. In: Kowalczyk, R., Huhns, M., Klusch, M., Maamar, Z., Vo, Q. (eds.) *International Workshop on Service-Oriented Computing: Agents, Semantics, and Engineering, SOCASE 2008*. LNCS, vol. 5006, pp. 56–68. Springer, Heidelberg (2008)
13. Jonquet, C., Dugenie, P., Cerri, S.A.: Agent-Grid Integration Language. *Multiagent and Grid Systems* 4(2), 167–211 (2008)
14. Wooldridge, M., Jennings, N.R., Kinny, D.: The Gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems* 3(3), 285–312 (2000)
15. Ferber, J., Gutknecht, O., Michel, F.: From agents to organizations: an organizational view of multi-agent systems. In: Giorgini, P., Müller, J.P., Odell, J. (eds.) *AOSE 2003*. LNCS, vol. 2935, pp. 214–230. Springer, Heidelberg (2004)
16. Dugénie, P.: UCS, Ubiquitous Collaborative Spaces on an infrastructure of distributed resources. PhD thesis, University of Montpellier (in French) (2007), <http://www.lirmm.fr/~dugenie/these>
17. Weiser, M., Brown, J.S.: Designing calm technology. *PowerGrid Journal* (July 1996)
18. Eisenstadt, M., Dzbor, M.: BuddySpace: Enhanced Presence management for collaborative learning, working, gaming and beyond. In: *JabberConf., Europe, Munich, Germany* (June 2002)

MASD: Towards a Comprehensive Multi-agent System Development Methodology

T. Abdelaziz^{1,2}, M. Elammari^{1,2}, and C. Branki³

¹IT Faculty, University of Garyounis, Benghazi, Libya

²University of Duisburg-Essen, ICB, (work was performed while being visiting researcher)

³School of ICT, University of the West of Scotland, Scotland, UK
tawill@garyounis.edu, elammari@garyounis.edu,
cherif.branki@uws.ac.uk

Abstract. In recent years multi-agent systems have gained a growing acceptance as a required technology to develop complex distributed systems. As result, there is a growing need for practical methodology for developing such systems. This paper presents a new Multi-Agent System Development (MASD) methodology, which has been developed over several years through analyzing and studying most of the existing agent-oriented ones. This methodology is constructed based on the strengths and weaknesses of existing methodologies. MASD aims to provide designers of agent-based systems with a set of methods and guidelines to allow them to control the construction process of complex systems. It enables software engineers to specify agent-based systems that would be implemented within an execution environment, for example Jadex platform. MASD differs from existing methodologies in that, it is a detailed and complete methodology for developing multi-agent systems. In this paper, we describe the process of the methodology illustrated by a running example namely a car rental system.

1 Background and Related Work

In the last few years, there has been an increasing acceptance of multi-agent system to implement complex and distributed systems. In addition, with the increasingly sophisticated applications being demanded by industry, agent oriented technologies is being supplemented. Building multi-agent applications for such complex and distributed systems is not an easy task. Indeed, the development of industrial-strength applications requires the availability of strong software engineering methodologies.

Many agent oriented methodologies and modeling languages have been proposed, such as: Gaia [18], MaSE [10], Tropos [4], HLIM [11], Prometheus [16], and AUML [2] etc. These methodologies are developed and specifically tailored to the characteristics of agents.

Many evaluation frameworks and comparisons of agent-oriented methodologies have been suggested such as [1] [3] [9]. Most of them agree on the fact that despite most of the methodologies are developed based on strong foundations, they suffer from number of limitations. These limitations have been stated as the following: none of the existing agent-oriented methodologies have been accepted as a standard and none of them is used and exploited in a wide manner [8]. Until now, no agent oriented

standards have been established, and research that examines and compares properties of these methodologies has suggested that none of them is completely suitable for industrial development of MAS [8]. Moreover, existing methodologies are based on strong agent-oriented basis, they need to support essential software engineering issues such accessibility and expressiveness which has an adverse effect on industry acceptability and adoption of agent technology [8]. Furthermore, up to this moment there is no well established development process to construct agent-oriented applications. Therefore, the consequences expected by the agents' paradigm cannot be fully achieved yet. None of them is in fact complete (in the sense of covering all of the necessary activities involved in software development life cycle (SDLC)) and is able to fully support the industrial needs of agent-based system development. Furthermore, most of the existing agent oriented methodologies suffer from the gap between design models and existing implementation languages [9]. One of the steps towards fulfilling this demand is to unify the work of different existing methodologies; work similar to development of the Unified Modelling Language in the area of object-oriented analysis and design.

In this paper, we propose a new methodology for multi-agent system development. The new methodology is developed based on the following main aspects of modelling techniques: concepts, models, and process.

2 MASD Methodology

MASD methodology is an attempt towards a comprehensive methodology through the unification of existing methodologies by combining their strong points as well as avoiding their limitations. It is developed as a reliable systematic approach that proves a milestone for Software Development Life Cycle (SDLC). The proposed methodology covers most important characteristics of multi-agent systems. It also deals with agent concept as a high-level abstraction capable of modeling complex systems. In addition, it includes well-known techniques for requirement gathering and customer communication and links them to domain analysis and design models such as UCMs [19], UML Use Case Diagrams, Activity diagrams, FIPA-ACL [13], etc. Furthermore, it supports simplicity and ease of use as well as traceability.

MASD methodology is composed of four main phases, System requirements phase, Analysis phase, Design phase, and Implementation phase. Fig. 1 illustrates the models of MASD Methodology. The next few sections present a more detailed discussion of each of the four phases. A car rental system [12] is used to describe the process of MASD methodology.

2.1 System Requirements Phase

The system requirements phase describes all the details of system scenarios as a high-level design through a system scenario model. The system scenario model uses well-known techniques such as Use-Cases Diagrams (UCDs) and Use Case Maps (UCMs) to describe the whole system scenario. Such techniques assist to discover system components such as (agents, objects, roles, resources ... etc.) and their high-level behaviour. The system requirements phase produces a model called system scenario model.

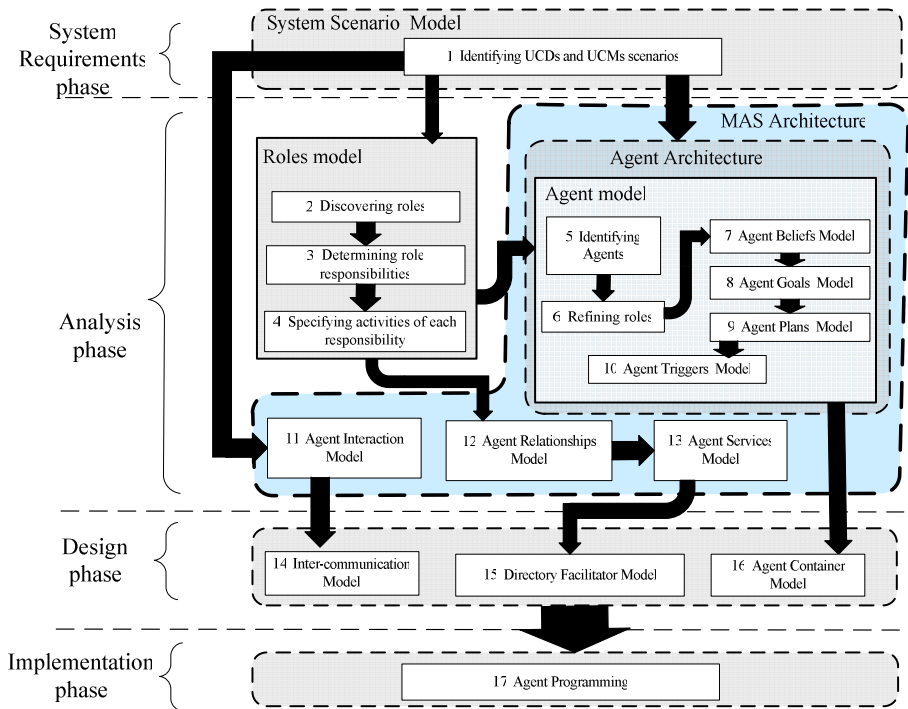


Fig. 1. MASD Methodology

This model is used as a starting point for generating more detailed visual descriptions. It describes the whole system scenarios in terms of what a system does but it does not specify how it does it. The model captures the components that the system is composed of and the tasks that have to be performed by each component within the system. At the system requirements stage, the system scenario model is developed by constructing UCDs as well as UCMs.

2.2 Analysis Phase

This phase is concerned with the description of the agent architecture as well as MAS architecture. It is divided into two stages. The first stage describes agent architecture. The second stage describes MAS architecture. Fig. 1 illustrates both architectures. The next sections provide a detailed description of both architectures.

2.2.1 Agent Architecture Stage

The agent architecture stage describes the following models: *Roles model*, *Agent model*, *Beliefs model*, *Goals model*, *Plans model* and *Triggers model*. MASD methodology requires the development of all models of the agent architecture stage. They are always developed even if the proposed agent system is just a single agent.

The agent role represents an agent behavior that is recognized, providing a means of identifying and placing an agent in a system. The agent can perform more than one role

in the system and more than one agent can perform the role. The roles as encapsulated units can be transferred easily from one agent to another when there is a need for that.

Roles model: discovers the roles that agents play or perform in the system, determines responsibilities for each role and specifies activities for each responsibility.

Agent model: identifies agents in the system and assigns roles to them, Refines the roles to fit agent capabilities. The agent model describes the internal structure of agents within the system and how these agents employ its internal knowledge and structure to perform their tasks. The agent model composed of two steps. Firstly, identify the agents within the system and refines the roles to fit agents' capabilities. Secondly, develop Beliefs model, Goals model and plans model as follows:

Agent Beliefs model: identifies agent beliefs. It stores relevant facts about the agent and its environment. Beliefs model in MASD is derived from the system scenario model and the roles model.

Agent Goals model: identifies the goals that the agent should achieve during the system run time. The goals model is captured from the role or roles that the agent plays in the system.

Plans model: specifies plans for each goal. It describes the plans that should be followed by an agent in order to achieve its goal.

Triggers model: identifies triggers that each agent should be aware of. Triggers could be events or change in beliefs.

2.2.2 MAS Architecture Stage

The MAS architecture stage includes the following models:

Agent interaction model: Identifies the interactions between agents in the system by using UCMs scenarios. These interactions explain the process in which agents exchange information with each other (as well as with their environment).

Agent relationships model: Captures the relationships between agents in the system to assist agents to identify dependencies between them.

Agent services model: Captures the services that each agent should provide in the system.

2.3 Design Phase

The design phase introduces the detailed representation of the models developed in the analysis phase and transforms them into design constructs. These design constructs are useful for actually implementing the new multi-agent system. The models that developed in the analysis phase are revised according to the specification of implementation. The main objective of the design phase is to capture the agent structural design and system design specifications. The design phase has three steps:

- i) Creating an agent container.
- ii) Constructing an inter-agent communications.
- iii) Creating a directory facilitator.

Design phase deals with the concepts that have been developed in analysis phase and illustrate how these concepts can be designed by identifying how to handle agent's beliefs, goals, and plans, as well as state how to compose the agent Capabili-

ties into reusable agent modules. Plus, it specifies the inter-communication among agents and how these agents are cooperated together to realize a common goal. A Directory Facilitator (DF) mechanism is also described.

2.4 Implementation Phase

Implementation phase is the point in the development process when we actually start to construct the solution and this is the time to start writing program code. During the implementation phase, the system is built according to specifications from previous phases. Previous phases provided models that can be transferred into implementation. . The produced models have a set of design specification showing how the agent system and its components should be structured and organized. The design specifications are used to develop the implementation phase. There are several agent frameworks and platforms proposed to develop multi-agent systems. MASD supports some of them, such as (JADE [14], JACK [7], MADKIT [15], Jason [4], and Jadexs [6]), as a tool to the development process.

3 Case study: Reservation Scenario for Car Rental System

Due to the restriction of paper size, we will describe the case study in brief.

3.1 System Requirements Phase (System Scenario Model)

The system scenario model describes UCDS and UCMS for reservation scenario of the car rental system in Fig. 2.

3.2 Analysis Phase

This phase describes two stages: agent architecture stage and MAS architecture stage.

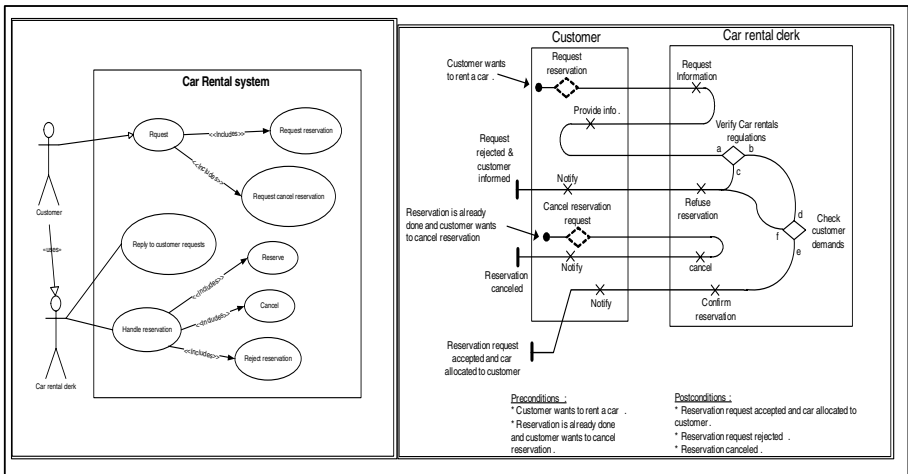


Fig. 2. UCD and UCM for Reservation scenario

3.2.1 Agent Architecture Stage

The agent architecture stage describes the following models:

Roles model: of the car rental system contains three roles (renter, rentier and director). Fig. 3 shows how the roles are extracted from UCMs and UCDs in the system scenario model. Fig. 4 illustrates the renter role that the customer component will play in the reservation scenario.

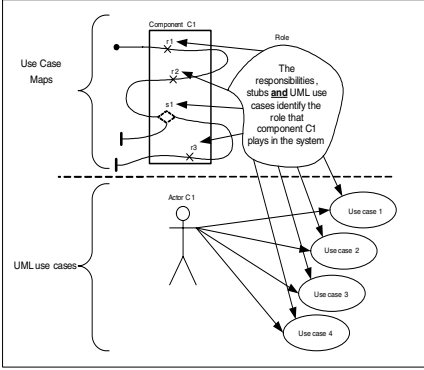


Fig. 3. Extracting roles from UCMs and UML use-cases

Role name	Renter
Role description	Renter who pays rent to use a car that is owned by the car rental company
Responsibilities & its Activities	Res: 1 Request reservation Act 1 Reserve car by a phone call Act 2 Reserve car by an E-mail Act 3 Reserve car by the Internet Res: 2 Cancel reservation request Act 1 Cancel reservation by a phone call Act 2 Cancel reservation by an Email Act 3 Cancel reservation by the Internet Res: 3 Notify real customer Act 1 Notify customer for cancelled reservations Act 2 Notify customer for rejected reservation Act 3 Notify customer for confirmed reservation
Obligations	<ul style="list-style-type: none"> The renter should pass rental regulation:
Permission	Null
Constraints	<ul style="list-style-type: none"> The renter should not have more than one reservation at the same time

Fig. 4. UCD and UCM for Reservation scenario

Agent model: identifies agents in the system and assigns roles to them, and refines the roles to fit agent capabilities. It is composed of two steps. Firstly, identify the agents within the system and refines the roles to fit agents' capabilities. The agent model contains three agents (customer, car rental clerk and car rental manager). Then assign the roles (renter, rentier and director) to the agents (customer, car rental clerk and car rental manager) respectively as shown in fig. 5. Secondly, develop the following sub-models: beliefs model, goals model, plans model and triggers model.

Agent Beliefs model: Describes the customer agent beliefs. It stores relevant facts about the customer agent and its environment. The beliefs model, as shown in Fig. 6, is derived from the system scenario model and the roles model.

Agent Goals model: Identifies the goals that the customer agent should achieve during the system run time. The goals of the customer agent, as shown in Fig. 7, are captured through the renter role that it plays in the system.

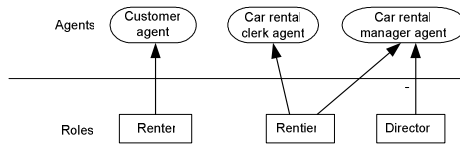


Fig. 5. Assigning roles to agents

Belief	Type	Purpos
Agent-4c	Constant	Storage
Customer wants to rent a car	Variable	Storage
Customer decides to reserve by phone	Variable	Storage
Customer decides to reserve by Email	Variable	Storage
Customer decides to reserve car online	Variable	Storage
Reservation confirmed	Variable	Storage
Reservation rejected	Variable	Storage
Customer wants to cancel reservation	Variable	Storage
Customer decides cancel reservation by phone	Variable	Storage
Customer decides cancel reservation by Email	Variable	Storage
Customer decides cancel reservation online	Variable	Storage
Cancellation confirmed	Variable	Storage
Reservation already done and car allocated to the customer	Variable	Storage
Cancel reservation is already requested by customer	Variable	Storage
The renter should fit to rental regulations	Variable	Storage
The renter should not have more than one reservation at the same time	Variable	Maintain

Fig. 6. Beliefs of the customer agent

Goal	Priority	Preconditions	Postconditions	Plans
Request reservation	High	Customer wants to rent a car	<ul style="list-style-type: none"> Reservation confirmed Reservation rejected 	<ul style="list-style-type: none"> Reserve car by phone call Reserve car by Email Reserve car online
Cancel reservation request	Normal	Customer wants to cancel reservation	<ul style="list-style-type: none"> cancelation confirmed 	<ul style="list-style-type: none"> Cancel reservation by phone call Cancel reservation by Email Cancel reservation online
Notify real customer	Normal	Real customer must be notified	<ul style="list-style-type: none"> Real customer must be notified 	<ul style="list-style-type: none"> Notify customer for canceled reservations Notify customer for rejected reservations Notify customer for confirmed reservations

Fig. 7. Goals of the customer agent

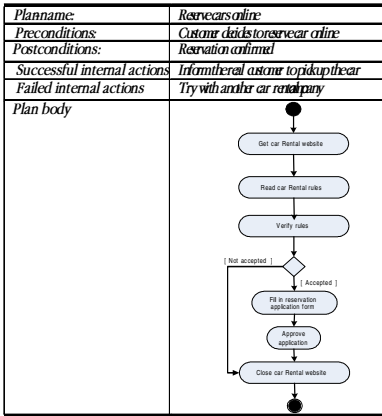


Fig. 8. Reserve online plan

Trigger name	Trigger type	Trigger activator	Actions
Customer wants to rent a car	Change of belief	Real customer	<ul style="list-style-type: none"> Request reservation (Goal)
Customer decides to reserve by a phone	Change of belief	Real customer	<ul style="list-style-type: none"> Reserve by a phone plan)
Customer decides to reserve by an E-mail	Change of belief	Real customer	<ul style="list-style-type: none"> Reserve by an E-mail plan)
Customer decides to reserve online	Change of belief	Real customer	<ul style="list-style-type: none"> Reserve Online plan)
Reservation confirmed	Event	Car rental clerk agent	<ul style="list-style-type: none"> Notify real customer to pick up the car plan)
Reservation rejected	Event	Car rental clerk agent	<ul style="list-style-type: none"> Notify real customer about a rejected reservation plan)
Reservation canceled	Event	Car rental clerk agent	<ul style="list-style-type: none"> Notify real customer about a canceled reservation plan)
Customer wants to cancel a reservation	Change of belief	Real customer	<ul style="list-style-type: none"> Cancel a reservation request (Goal)
Customer wants to cancel a reservation by a phone	Change of belief	Real customer	<ul style="list-style-type: none"> Cancel a reservation by phone plan)
Customer wants to cancel a reservation by an E-mail	Change of belief	Real customer	<ul style="list-style-type: none"> Cancel a reservation by an E-mail plan)
Customer wants to cancel a reservation online	Change of belief	Real customer	<ul style="list-style-type: none"> Cancel a reservation Online plan)
Cancellation confirmed	Change of belief	Car rental clerk agent	<ul style="list-style-type: none"> Inform a real customer plan)

Fig. 9. Triggers of the customer agent

Plans model: Fig. 8 describes *Reserve online* plan for *Request Reservation* goal. It describes the plan preconditions that should be satisfied in order to enable the agent to achieve the *Request Reservation* goal.

Triggers model: Identifies the triggers that customer agent should be aware of as being events that take place in the system.

3.2.2 MAS Architecture Stage

The agent architecture stage describes the following models:

Agents Interaction Model: Identifies the interactions between customer agent and car rental clerk agent in the system by using UCMs scenarios. Fig. 10 describes how the interactions between customer agent and car rental clerk agent are derived from UCMs scenarios.

Agent relationships model: describes relationships between the customer agent and the car rental clerk agent as shown in fig. 11.

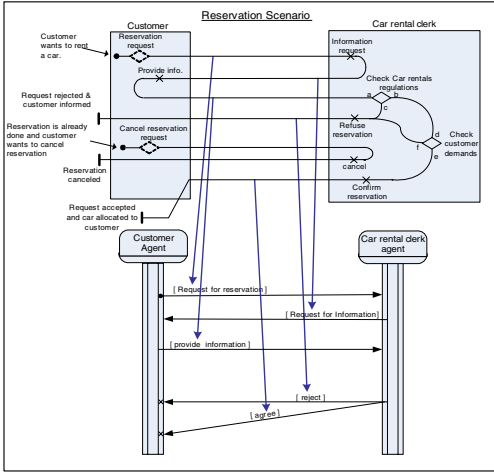


Fig. 10. Mapping from UCMS scenarios to interaction diagrams

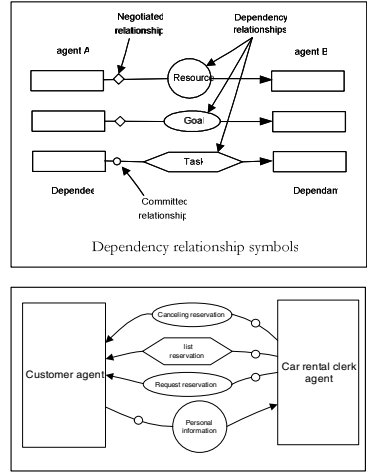


Fig. 11. Dependency diagram

Agent service model: is derived from the use-case diagrams that were developed in the system scenario model. Agent services can be derived directly from use case diagrams where each use case can be identified as service. Fig. 12 illustrates the agents' services model including the car rental clerk agent services.

Service	Agent	Expiry date	Time of availability	Cost
Reply to customer inquiries	Car rental agent	Open	always	Free
Handle reservation request	Car rental agent	Open	8:00 am to 8:00 pm	Free
Handle rental	Car rental agent	Open	8:00 am to 8:00 pm	Free
Handle car service	Car rental agent	Open	8:00 am to 4:00 pm	Free

Fig. 12. Agent services model

3.3 Design Phase

The design phase has three steps: firstly, creating an agent container. Secondly, constructing an inter-agent communications. Finally, creating a directory facilitator.

Agent container: In the agent container, all the models that have been developed in the agent architecture stage are revised and extended by some fields to fit the implementation environment. We have chosen Jadex for the implementation environment.

Inter-agent communication: this step transforms the interaction diagrams that developed in the MAS architecture stage into FIPA ACL protocols.

Directory facilitator: the agent services model and agent relationships model that developed in the MAS architecture stage are extended into directory facilitator according to the specification of Jadex platform.

3.4 Implementation Phase

In this phase, design models are transformed into code. We used Jadex platform to implement the case study. Jadex was chosen because it is a Java based, FIPA compliant agent environment, and allows developing goal-oriented agents following the BDI model.

4 Conclusions and Future Work

Agent oriented approaches represent an emerging paradigm in software engineering. Therefore, the availability of agent oriented methodologies that support software engineers in developing agent based systems is very important. In recent years, there have been an increasing number of methodologies developed for agent-oriented software engineering. However, none of them are mature and complete enough to fully support the industrial needs for agent based system development.

For all those reasons, it is useful to begin gathering together the work of various existing agent oriented methodologies with the aim of developing a new methodology and a step towards a comprehensive one. Thus, this research paper is focused on developing a design methodology to assist a multi-agent system designer through the entire software development lifecycle, beginning from system requirement phase, and proceeding in a structured manner towards working code.

There are few principal strengths of the methodology developed through this research work. First, it is based on three important aspects: concepts, models, and process, and it is focused toward the specific capabilities of multi-agent systems. At the commencement of research, MASD combined several techniques and concepts into a single, simple, traceable, and structured methodology. These concepts and techniques are represented through a set of models. Most of these models used within the methodology have therefore been already justified and validated within the domain of agents and multi-agent systems. MASD provides an extensive guidance for the process of developing the design and for communicating the design within a work group. It was very clear that the existence of this methodology provides a great assistance in thinking about and deciding on the design issues, as well as conveying design decisions.

With regards to future work, there are some topics that need to be investigated. Topics such as: domain specific issues, testing of the resulting software, and agent project management (metrics, estimation, schedule, risk, quality).

References

1. Abdelaziz, T., Elammari, M., Unland, R.: A Framework for the Evaluation of Agent-oriented Methodologies. In: 4th International Conf. on Innovations in Information Technology, Dubai, UAE (2007)
2. Bauer, B., Odell, J.: UML 2.0 and agents: how to build agent-based systems with the new UML standard. *Journal of Engineering Applications of Artificial Intelligence* 18(2) (2005)

3. Bobkowska, A.E.: Framework for methodologies of visual modeling language evaluation. In: Proceedings of the symposia on Meta-informatics, ACM Press, New York (2005)
4. Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J., Perini, A.: TROPOS: An Agent-Oriented Software Development Methodology. *Journal of Autonomous Agents and Multi-Agent Systems* 8, 203–236 (2004)
5. Bordini, R.H., Hübner, J.F., et al.: Jason, manual, release 0.7 edition (August 2005), <http://jason.sf.net/>
6. Braubach, L., Pokahr, A., Lamersdorf, W.: Jadex: A Short Overview. In: Main Conference NetObjectDays, Erfurt, Germany (2004)
7. Busetta, P., Rönquist, R., Hodgson, A., Lucas, A.: JACK Intelligent Agents Components for Intelligent Agents in Java. Updated from AgentLink Newsletter (October 1999), <http://www.agent-software.com.au/>
8. Dam, K.H., Winikoff, M.: Comparing Agent-Oriented Methodologies. In: Giorgini, P., Henderson-Sellers, B., Winikoff, M. (eds.) AOIS 2003. LNCS (LNAI), vol. 3030, Springer, Heidelberg (2004)
9. Dastani, M., Hulstijn, J., Dignum, F., Meyer, J.: Issues in Multi-agent System Development. In: AAMAS (2004)
10. DeLoach, S.A.: The MaSE Methodology. In: Methodologies and Software Engineering for Agent System. The Agent-Oriented Software Engineering Handbook Series, vol. 11 (August 2004)
11. Elammari, M., Lalonde, W.: An Agent-Oriented Methodology: High-Level and Intermediate Models (HLIM). In: Proceedings of AOIS, Heidelberg (1999)
12. EU-Rent, EU-Corporation, <http://www.businessrulesgroup.org/egsbrg.shtml>
13. JADE: Java Agent Development Framework (1999), <http://jade.csel.tu.it>
14. MADKIT: Multi-Agent Development KIT (1999), <http://www.madkit.org>
15. Padgham, L., Winikoff, M.: Prometheus: A methodology for developing intelligent agents. In: Third Workshop on Agent-Oriented Software Engineering (2002)
16. Sturm, A., Dori, D., Shehory, O.: Single-Model Method for Specifying Multi-Agent Systems. In: The Second International Joint Conference on Autonomous Agents and Multi-agent Systems, July 14-18, Melbourne, Australia (2003)
17. Zambonelli, F., Jennings, N.R., Wooldridge, M.: Developing multi-agent systems: The Gaia methodology. *ACM Transactions on Software Engineering and Methodology* 12(3), 317–370 (2003)
18. Buhr, R.J.A.: Use Case Maps as Architectural Entities for Complex Systems. *IEEE Transactions on Software Engineering* 24(12), 1131–1155 (1998)

A Mobile Device Based Multi-agent System for Structural Optimum Design Applications

Cherif Branki, Tilmann Bitterberg, and Hanno Hildmann

University of the West of Scotland, PA1 2BE Paisley, Scotland, UK
{cherif.branki,georg.bitterberg,hanno.hildmann}@uws.ac.uk

Abstract. This paper will discuss a design, implementation and evaluation of a multiagent system (MAS) for a structural optimum design application (SODA) on mobile devices. The paper commences by defining the relevant concepts of SODAs and MAS. It will then present an architecture and framework combining both. It will explore in detail an implementation running on a mobile device and compare that to a version on a standard PC. The computational evaluation results obtained from a simulation on an actual mobile device indicate that running a complex system on such a device is feasible and usable. Finally, the paper suggests that current mobile devices can be used as more than data displays, and their processing power can be efficiently utilised.

1 Introduction

In recent years, mobile phones received an increasing attention from various scientific communities due to their pervasive nature and because the standard products today are essentially small, mobile computers. Hence it makes them a tool worthy of consideration, especially for applications where the user cannot be expected to purchase an expensive interface or where mobility is of importance.

In this paper a design and implementation of a multi-agent system for structural optimum design on mobile devices is reported. The application presented is the result of a research project on automated structural design, more specifically a tool to assist in the design of power cable trusses (steel constructions of sizeable dimensions that are used extensively for the delivery of electrical power to households). Originally the application was conceived as a tool to be run in advance of a construction as an advisory to the structural designer. Preliminary results of the existing PC based implementation have however suggested that the application can be extended to server as an *on site* tool to render assistance ranging from providing estimates on material requirements to feasibility studies for small and medium sized construction projects.

2 Structural Optimum Design Applications (SODAs)

Traditionally structural optimum design was primarily concerned with searching for both the optimum properties for individual structural components (topology)

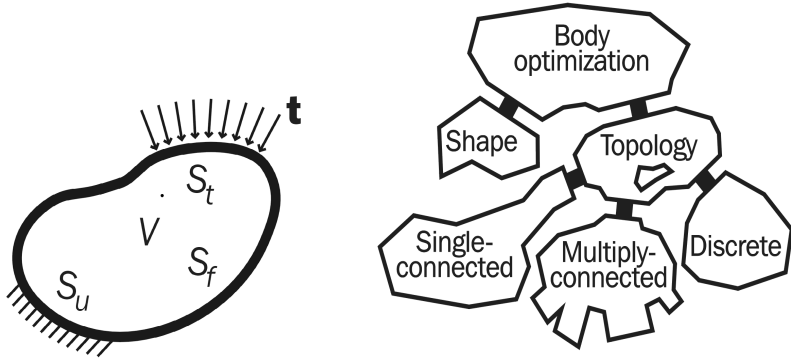


Fig. 1. A deformable body (left), the topology optimisation problem (right)

as well as the optimisation of the shape of the overall structure (geometry). Recent research attention has shifted towards topological optimisation. In this section we introduce a general formulation of *optimum design*, more specifically of *optimum design in mechanics of continuum* [2].

Definition 1 (A deformable body). Let a deformable body of volume V be enclosed by a surface S . The part S_t of this surface is subject to traction t , the part S_u is fixed and the remaining part S_f is free. See Figure 1 for an illustration.

Generally, optimisation problems for bodies described in Definition 1 can be distinguished as belonging to either of the following categories: *Shape optimisation* (See Definition 2) and *topology optimisation* (See Definition 3).

Definition 2 (Shape optimisation). Let a deformable body of volume V be enclosed by a 3 dimensional environment with at least two defined points p_1, \dots, p_i . Furthermore assume that the body is required to occupy all i points in space simultaneously. Assuming there is more than a single solution to this problem then it can be subjected to a number of requirements with respect to the spacial coordinates (other than the i points) occupied by the body.

The *requirements* mentioned in the Definition above can range from aesthetic considerations to practical, physical or spacial constraints. In our case and the presented case study the requirements are spacial and physical in nature.

Since we can take the material of the body as a constant when searching for the optimum shape we can concentrate on finding the best configuration for the free surface S_f making *shape optimisation* the easier one of the two categories.

Definition 3 (Topology optimisation). Let $P = \{p_1, \dots, p_i\}$ and $Q = \{q_1, \dots, q_j\}$ be disjoint set of spacial coordinates. Let furthermore V be a deformable body occupying all points in P and none in Q . Topological optimisation is then the best possible arrangement of V over the remaining spacial coordinates.

Regarding topological optimisation we consider three alternative approaches:

1. We can simplify our representation of the body in question by treating it as one single-connected region that is filled with a material (of variable density). On the assumption that the total amount of material at our disposal is fixed the problem then becomes one of distributing the available material within the body [3,4].
2. By introducing cavities into body V we can focus on optimising their shape as well as position and number within the constraints set by the properties of the surrounding material [5]. This approach assumes that the connectivity of body V is larger than 1.
3. Regarding the granularity of the internal structure there are a multitude of intermediate views that would fall in between approach [1] and [2], for example composites, periodic non-homogeneity, etc. As a third approach we can abandon the continuum assumption altogether and model our body V as a finite number of elements connected to one another, i.e. as a discrete structure. The MAST project described as a case study in Section [4] uses this conceptual model.

In addition to the two categories for optimum design there is one more aspect to consider: The components from which to construct the structure. This is the third conceptually different aspect to consider for the optimum design problem. We argue that a brute force attempt to solve the problem does very quickly become too expensive in terms of complexity and computation time. By breaking the problem down over these 3 levels we reduce the complexity and allow for intermediate verification of our solution (i.e. checking for feasibility and allowing backtracking to a previous level if necessary). Figure [2] shows the relation between these 3 levels. We explain this below when we elaborate on the individual levels in more detail. The initial considerations are on the topological level and can be stated as two questions:

1. What are the individual building blocks of which the physical structure is composed? Here we are not concerned with the properties of individual components but consider only the different types at our disposal.

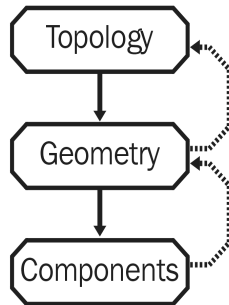


Fig. 2. The three levels of an optimum design task

2. In which manner can the different types of building blocks be connected to one another? Again we are considering this on a high level, i.e. we are considering whether two types of building blocks can at all be joined, not how they should be joined for that is done on the geometric level.

At the geometric level we are considering the question of which individual blocks to use and how these block are connected to one another in order to meet the specifications for the complete structure. This explains the dotted arrows in Figure 2 (leading back from the geometric level to the topological one) because if the problem can not feasibly be solved with the block available to us then we have to backtrack to the previous level and reconsider our choices there.

The last step after determining the order in which the building blocks are to be assembled to meet the overall design specifications is to investigate the requirements each of these blocks now has to meet. Consider for example a long horizontal truss constructed of a single type of building block and attached to a wall on only one side. Clearly the forces that act upon these identical block will be larger the closer we move to the block attached to the wall (*Law of the lever*). Due to this solutions could be drastically improved by replacing a block with one of a larger structural strainability. Alternatively, the requirements for the individual building block of a geometrically sound construction might exceed safety limits, forcing the designed to backtrack to the geometric level to construct an alternative (geometric) solution.

This 3-level approach is normally not used in practise. The designed often has no influence on the types of building blocks available and is only rarely in the position to request individual building blocks with specific properties. Due to this there are many structures (e.g. power line masts) that are mass produced to meet a set of realistic requirements, resulting in many of them being much stronger (e.g.) than they would need to be, and subsequently, more costly.

3 Mobile Devices

The Gartner research institute stated that global sales of mobile phones (in 2006) were in excess of 0.9 billion units [6]. A more recent quote (dated May 28th) states that “*Worldwide sales of mobile phones reached 294.3 million units in the first quarter of 2008, a 13.6 per cent increase over the first quarter of 2007*”.

However when considering the whole range secondary products (handsets, internet connection, applications, etc) the magnitude of this still very young industry becomes clear. As estimated by the Juniper research organization, the market for digital mobile phone products alone (mobile games, TV, music, adult content, etc) will reach a volume \$47 billion in 2009 and, looking further into the future, \$77 billion in 2011 [6]. This undeniably huge industry and the already pervasive status of mobile technologies, especially mobile phones, has shaped our society as a whole and has become an integral element of every day life for the general public including senior citizens, young adults and even children.

¹ <http://www.gartner.com/it/page.jsp?id=680207>, as verified on August 24th, 2008.

In recent years mobile phones have received an increasing attention from the scientific community both as devices of notable computational power [7] as well as in their capacity as nodes in distributed computing applications [8]. [9] argues that the multi agent paradigm is an ideal middleware for e.g. the management of mobile devices. The distributed problem solving approach underlying the multi-agent system idea is especially relevant for scenarios where the effective use of (limited) resources is of high importance, which supports the suggestion to use it for mobile phone based applications as much as the claim that multi agent systems are likely to perform well for SOD problems.

This is our motivation to use this technology both as platform for content applications in general as well as terminals to interact with server based applications that require more computational power than a mobile device can currently provide. In recent work [10,11,7,12] we have already argued for the use of mobile technologies and pointed out that due to the computational power of contemporary mobile phones complex applications can now feasibly be implemented on devices that have long since become common household items.

4 Case Study: The MAST Project

In this section the following example illustrates the concepts and operations of the MAST architecture.

Lets consider a (human) structural designer (SD) who wants to test out a new configuration of a truss. The SD already has a vague idea of how the truss should look like but effectively the SD wants to try different things until the SD finds the most suitable solution.

4.1 The MAST Project

In Section 2 we briefly discussed three approaches to the problem of topological optimisation that the MAST project adopts. This decision is motivated by our belief that an initial discrete description is beneficial. Amongst other reasons because it does not force us to reproduce the discrete optimal structure artificially, as the continuous model would.

The MAST project is ongoing research and it's final goal is to develop a Multi-Agent Structural optimisation Tool (MAST). Contrary to existing SOD-packages, the results of this research work will perform the search for optimum solution at three levels: the topological level, the geometrical level and the level of components. The common criterion of minimum cost (weight) will be expanded by two criteria related to the vulnerability of structure to the terrorist attack.

The architecture of the MAS is heavily influenced by the internal representation of the data structure respectively the data inside the already available prototype software. The representation chosen is strongly coupled with the application itself thus making it difficult to design and implement the MAS on top of this. The prototype (MAST) is also monolithic meaning there is no easy way of pulling and pushing of data in and out which is a fundamental requirement

for the MAS. A solution to overcome this limitation is presented in [13] which has been proven to be successful.

4.2 MAST Architecture

In the architectural overview (presented in [2]) there are three main components of the system; the Panel Designer, the MAST Prototype Application and the Communication Framework. The Self-Organisational Module presented in [2] is an optional plug-in and does not form part of the scope of this paper.

The Panel Designer is a graphical user interface tool to design structural elements. This is a very advanced and convenient tool to create new structural elements and to edit existing ones. The tool is able to read and write the same data files as all other modules in the system. The Data Converter in between the Panel Designer and the MAST Application is a simple module which translates geometrical data into a normalised form. The Panel Designer is currently not available for mobile devices.

The MAST Application encapsulates the above mentioned optimum structural design. It implements the technologies by constructing trusses and cantilevers using a limited alphabet of structural elements, the so-called panels. It uses hierarchical graph grammars and evolutionary algorithms to find an optimum structural design by generating a topological and a geometrical optimisation.

The Communication Framework was designed as such to extract data in and out of the MAST Application and into the project. It is the backbone of the whole system providing data channels to various applications.

The framework is built on a distributed system, so the various applications using the framework do not necessarily have to be executed on the same physical system. It features a so-called registry which enables the hook up of plug-ins and small programmes.

The MAS is designed to utilise the Communication Framework to retrieve, store and evaluate data and the agents use the Communication Framework to negotiate their results.

5 Computational Results

The system outlined in the previous sections has been developed and implemented. We choose Java as its implementation environment because of the language's cross-platform properties and its availability on nearly all modern mobile devices. For the case-study, an optimising application has been developed on NetBeans 6.1 and is to be run on a regular personal computer and on a mobile device. The JavaVM used to execute the application on a PC is a current Java SE 1.6. On the mobile device the Sun Java(TM) Wireless Toolkit 2.5.2 for CLDC is used. The mobile development tool chain provided by NetBeans is excellent making development, testing and actual deployment an ease.



Fig. 3. Two input screens of the implementation for the (emulated) mobile device

5.1 Parameters

The input parameters for the application are chosen in a way to actually produce a useful result i.e. a good optimum design is obtained. The parameters remain the same for all evaluations to follow to be able to compare not only the results but also the performance of the actual calculation itself. There are four main parameters (see left hand image in Figure 3).

1. Topology/Geometry: Controls where the main focus of this optimisation should lie on. The agents behave very similar for any of these two choices yet there are slight differences in the amount of calculations involved.
2. Initial Population: The amount of initial random structures. The more the better but affects performance. We have found that a value of 100 is sufficient. A higher value while possible improving the result, will only do so marginally and at an added cost that does not warrant the additional memory needed for the operation.
3. Generations: The amount of negotiation rounds between agents. More is better for the result but increases runtime in a linear way.
4. Environment: This setting defines the initial geometric constraints (see right hand image in Figure 3). Ideally, these constraints will originate from a picture taken with the phone's camera. We present a list of common landscape profiles each of which will influence the agents in their negotiations in a different way.

5.2 Facts and Figures

With these parameters set, we continue to profile the application for CPU and Memory usage. As outlined above, the more objects are involved (Population) the more memory is required; the more negotiation rounds are run (Generations),

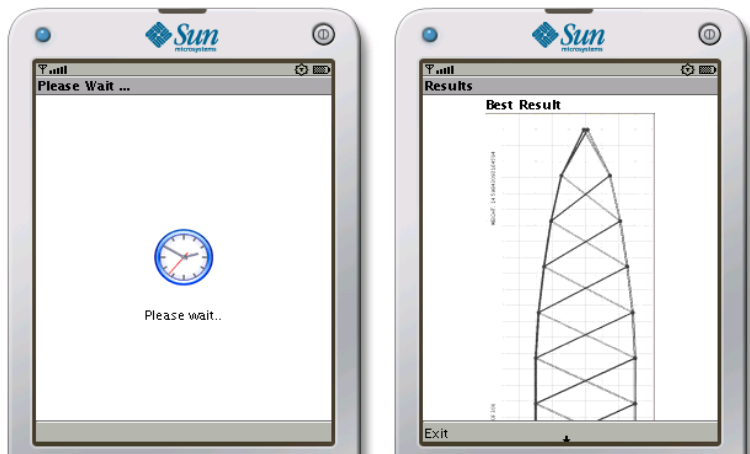


Fig. 4. Screenshots of some output screens of the (emulated) mobile device

the more CPU is needed. A single run of the application produces over four million method invocations resulting from calling 161 different methods.

The four computationally most expensive routines are invoked 200.000 times and account for roughly 10 percent of total execution time the remainder being user interface code paths, initial initialisation of the application and results display. Memory wise, the application requires a little bit more than 3MB of main memory during its execution. The MAS needs a maximum of 1MB during the negotiation rounds. The compiled binary is just 176KB small.

5.3 Results for: PC, Emulated Devices and Mobile Phones

On a current personal computer the execution time for the application configured with the above mentioned parameters is less then two seconds and therefore barely noticeable and most acceptable to the user. The user interface is by far the most easiest one to learn and use.

For preliminary feasibility studies and to assess realistic scope of the project an intermediate implementation was prepared and tested in a mobile phone emulator. Inside the mobile phone evaluator provided by NetBeans the total execution time increases to 5 seconds. Since the MAS and the infrastructure of the mobile version is exactly the same as in the PC version this already suggests that running the MAS on a mobile device is indeed a possibility. The user interface had to be completely rewritten and does no longer have any resemblance with the PC version. By abstracting various parameters and accommodating for the limited display size of a mobile device the application retains its full usability.

Running the application on an actual mobile phone only slightly increases the total execution time by another 40 percent from the emulator time pushing the total to 9 seconds. This is still an acceptable time as most mobile phones take longer to boot up or start an internet browser.

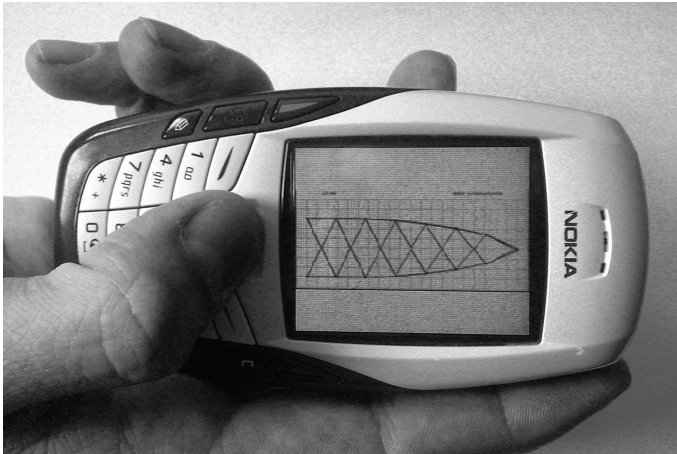


Fig. 5. Screenshot of the MAST application running on a mobile phone

6 Conclusion

We have argued in the sections above that SODAs can be successfully combined with the MAS. The flexibility of a MAS adds value to a SODA as it adapts quickly to new and unknown external constraints onto the system. The computational results presented support this claim.

Furthermore, the application on the PC could easily be geared towards a web-service or made network-accessible in a similar way. However solving the problem externally will take longer than the advised time from above. Finally recent mobile devices shall not just be used as dumb terminals for data display only as they can perform very complex operations, and their processing power and memory shall be utilised. Porting stand-alone algorithms from one java platform to the other in order to become mobile is a solvable problem and should be considered for specialised applications for mobile devices.

References

1. Biegus, L., Branki, C.: India: a framework for workflow interoperability support by means of multi-agent systems. *Engineering Applications of Artificial Intelligence - Autonomic Computing Systems* 7(17), 825–839 (2004)
2. Bitterberg, T., Branki, C., Borkowski, A., Grabska, E.: Self-organizing agents approach to structural design. *International Transactions on Systems Science and Applications* 1(3), 278–282 (2006)
3. Bendsoe, M.: *Methods for the Optimisation of Structural Topology, Shape and Material*. Springer, Berlin (1995)
4. Bletzinger, K.U., Kimmich, S., Ramm, E.: Efficient modeling in shape optimal design. In: Topping, B.W.E. (ed.) *Computational Structures Technology*, pp. 1–15. Herriot-Watt University, Edinburgh (1991)

5. Eschenauer, H., Schumacher, A.: Bubble method for topology and shape optimisation of structures. *Journal of Structural Optimisation* (8), 42–51 (1994)
6. Stenbacka, B.: The impact of the brand in the success of a mobile game: comparative analysis of three mobile j2me racing games. *Comput. Entertain.* 5(4), 1–15 (2007)
7. Bitterberg, T., Hildmann, H., Branki, C.: Using resource management games for mobile phones to teach social behaviour. In: *Proceedings of Techniques and Applications for Mobile Commerce (TAMoCo 2008)*, Glasgow, Scotland, pp. 77–84. IOS Press, Amsterdam (2008)
8. Kurkovsky, S., Bhagyavati, Ray, A.: A collaborative problem-solving framework for mobile devices. In: *ACM-SE 42: Proceedings of the 42nd annual Southeast regional conference*, pp. 5–10. ACM, New York (2004)
9. O’Sullivan, T., Studdert, R.: Agent technology and reconfigurable computing for mobile devices. In: *SAC 2005. Proceedings of the 2005 ACM symposium on Applied computing*, pp. 963–969. ACM, New York (2006)
10. Meissner, A., Baxevanaki, L., Mathes, I., Branki, C., Schönfeld, W., Crowe, M., Steinmetz, R.: Integrated mobile operations support for the construction industry: The cosmos solution. In: *Proceedings of of the 5th World Multi-Conference on Systemics, Cybernetics and Informatics, SCI 2001*, pp. 248–255. International Institute of Informatics and Systemics, IIIS, Orlando, FL (July 2001)
11. Meissner, A., Mathes, I., Baxevanaki, L., Dore, G., Branki, C.: The cosmos integrated it solution at railway and motorway construction sites - two case studies. *ITcon, Special Issue eWork and eBusiness* 8, 283–291 (2003)
12. Hildmann, H., Uhlemann, A., Livingstone, D.: A mobile phone based virtual pet to teach social norms and behaviour to children. In: *Proceedings, The 2nd IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning (Digitel 2008)*, Banff, Canada (November 2008)
13. Branki, C., Bitterberg, T., Bridges, A., Borkowski, A., Grabska, E.: Three layered agents evaluators for three layered structural optimisation problems in a multi agents structural tool. In: *Proceedings of EuropIA*, vol. 10, pp. 239–249 (September 2005)

Discovering Pragmatic Similarity Relations between Agent Interaction Protocols

Maricela Bravo¹ and José Velazquez²

¹Morelos State Polytechnic University, Cuauhnhuac 566, Texcal,
Morelos, México, CP 62550
mbravo@upemor.edu.mx

²Electrical Research Institute, Reforma 113, Palmira,
Morelos, México, CP 62490
jconrado@iie.org.mx

Abstract. A multi-agent system (MAS) consists of a set of autonomous agents, capable of interacting among each other with cooperation or coordination purposes. To achieve their goals, all agents in a MAS must exchange messages following a interaction protocol. Currently there are research efforts to provide standard mechanisms to achieve cooperation between multiple heterogeneous MAS. However, there are still some issues that must be solved in order to fully automate integration and inter-operation among them. In this paper we present a novel approach for discovering pragmatic similarity among multiple agent interaction protocols, our objective is to support system developers and integrators generating a set of pragmatic relations between pairs of agent interaction protocols and a numerical measure which represents the similarity among them. We describe an example to show the applicability of our approach.

Keywords: Pragmatic similarity, multi-agent systems, interaction protocols.

1 Introduction

A MAS consists of a set of autonomous, independently developed agents, which have interaction capabilities for communication, negotiation, cooperation and coordination among them. To achieve their goals all agents in a MAS, exchange messages following a communication protocol. Currently Internet-based environments have gained more attraction than ever, and many agents independently developed are being deployed on the Web, causing the problem of heterogeneity. Heterogeneity in MAS has many causes: use of different hardware and operating systems, use of different programming languages for implementation, different data base management software, different naming techniques, different data and code formats, etc. Considering this inherent MAS heterogeneity, to provide agents with mechanisms to be adaptable at run time to interoperate in a dynamic and complex environment such as Internet, represents an open problem. Therefore, research efforts to discover similarity relations among different agent interaction protocols, preserving their autonomy and with no reprogramming requirements will benefit the automatic interaction of MAS over Internet.

Many authors have approached this problem from the Web Semantic perspective, by using taggers, classifiers, online dictionaries, ontologies, among others. We consider this is a good solution approach, which has proven good results. However, this semantic approach has been mainly used for heterogeneous information sources integration: such as distributed data bases, vocabularies, ontologies, taxonomies, etc. This semantic approach would be enough if Internet was populated only with information sources, but it is not. Indeed, Internet has more than only data; it has programs, agents, Web applications, Web services, etc. We will refer to this kind of sources as interactive software agents. These interactive agents have to interoperate with each other through the exchange of messages, following interaction protocols or rules.

In this paper we focus on the pragmatic aspect of interactive agents. Pragmatic similarity represents a binary relation between two issued messages if their use is of similar intention. In particular we propose an approach for computing pragmatic and generating similarity relations between agents, which use different interaction protocols. Our approach is based on the analysis of transition functions from Finite State Machines (FSM). The aim of this research is to provide software programmers or system integrators with mechanisms to compute pragmatic similarity among interactive agents.

In contrast to the semantic approach, which works with data sources, we propose a different approach for comparing software pragmatics based on their message passing logics. The result of this analysis helps the developer to measure pragmatic similarity, in order to design and implement a better solution. For example, translation, learning or reprogramming all agents to be fully interoperable among.

The rest of the paper is organized as follows. In section two we present a brief description of representation formalisms which have been used for modeling, simulating and implementing interactions in MAS. In section three we describe the process for discovering pragmatic similarity relations. In section four we present an example to show the applicability of our approach, and finally, in section five we conclude.

2 Pragmatic Representation Formalisms

There are various formalisms reported in literature to model MAS, for example Petri Nets, Colored Petri Nets, Pi-calculus, AUML, BPEL, OWL-S.

Petri Nets. Petri Nets were first introduced by Carl Adam Petri, as a result of his Ph.D. thesis in 1962 [1]. Petri Nets have been used to analyze and verify systems in different areas of science, such as artificial intelligence, concurrent systems, control systems, analysis of networks, etc. Petri Nets represent a traditional formalism for modeling interactions and concurrency. A Petri Net is a directed, connected, bipartite graph with annotations, in which each node is either a place or a transition. Tokens are in places, when there is at least one token in every place connected to a transition, then that transition is enabled.

Colored Petri Nets. Colored Petri Nets [2] are based on Petri Nets, but they have added properties. Tokens are not simply blank markers, but have data associated to them. A color in a token represents a schema or type specification. Places are sets of tuples, called multi-sets. Arcs specify the schema they carry, and can also specify

Boolean conditions. Arcs exiting and entering a place may have an associated function which determines what multi-set elements are to be removed or deposited. Boolean expressions, called guards, are associated with the transitions, and enforce some constraints on tuple elements. Colored Petri Nets are equivalent to Petri Nets, but the richer notation of colored Petri Nets makes them more suitable for modeling interactions with more information.

Pi-Calculus. Is a process algebra presented in [3]. It is a formalism for modeling concurrent processes, whose configurations may change as the process executes over time. In Pi-Calculus the fundamental unit of computation is the transfer of a communication link between two processes. The simplicity of the Pi-Calculus is because it includes only two kinds of entities: names and processes. These entities are sufficient to define interaction behavior.

AUML. Agent Unified Modeling Language is a representation formalism which facilitates the visual development of multi-agent systems, with emphasis on agent conversations. It was first introduced by [4]. AUML is concerned mainly with interaction diagrams for conversation modeling. Interaction diagrams mainly extend the OMG definition of UML sequence diagrams with the possibility to express explicitly concurrency in the sending of messages.

BPEL. Business Process Execution Language [5] is a formalism used for specifying the composition of Web services. It was created to standardize interaction logic and process automation between Web services. BPEL is a convergence of language features from WSFL and XLANG. However, this language lacks well defined semantics, which makes it difficult to reuse and compose.

OWL-S. OWL-S [6] is one of the standards for the description of Web services. It includes a process model for Web services. Each process is described by three components: inputs, preconditions and results. Results specify what outputs and effects are produced by the process under a given condition.

FSM. Finite State Machine [7] represents a powerful formalism for describing and implementing the control logic of an interaction system. They are suitable for implementing communication protocols, control interactions and describe transitional functions. FSM mainly consist of a set of transition rules. In the traditional FSM model, the environment of the machine consists of two finite and disjoint sets of signals, input signals and output signals. Also, each signal has an arbitrary range of finite possible values.

The above are all good representation formalisms to model interactions among distributed agents. However, not all have the same purpose. Some are good for modeling interactions formally (Petri Nets, Colored Petri Nets, Pi-Calculus and AUML), others have tools for verification and simulation (Petri Nets and Colored Petri Nets), others are good for executing and implementing composite processes (BPEL and OWL-S). But for the aim of this work we have selected FSM because they offer a simple manner of implementing transitional functions in order to compute similarity in pragmatics of represented protocols.

3 Discovering Pragmatic Similarity

For discovering pragmatic similarity relations, interaction protocols between agents should be represented in a computable formalism. As it was described in Section 2, there are various formalisms reported to achieve this goal. However, some are useful for modeling, some others are good for executing processes, some are good for simulating interaction processes, but we needed to use a formalism easy to implement and therefore to compute. We also needed a formalism which would help us in the automatic discovering of functionalities, which is our work in progress. Thus, we selected FSM for this reason.

A **FSM** is a tuple $(S, I, O, ft, fo, s0)$, where

S is a finite set of states,

I is the set of inputs,

O is the set of outputs,

ft is the transitional function,

fo is the output function and

$s0$ represents the output state.

A **transition function** is represented by $ft = (is, im) = fs$, where

is is the initial state,

fs is the final state and

im is the input message.

For the purpose of our work we have adapted the machine equivalence definition to define a pragmatic equivalence on their transition functions.

Machine equivalence. Let $M = (S, I, O, ft, fo, s0)$, and $M' = (S', I', O', ft', fo', s0')$ be two FSM. States $s \in S$ and $s' \in S'$ are equivalent if their transition functions ft and ft' coincide.

Pragmatic equivalence relation. Let ft be a transition function of M and ft' a transition function of M' . Input messages $im \in I$ and $im' \in I'$ are equivalent if their initial state $is \in S$ is equal to $is' \in S'$, and if their final states $fs \in S$ is equal to $fs' \in S'$ are equal.

In order to compare pragmatic similarity among different interaction protocols we established a common set of states for all protocols, to allocate all the messages and transition functions in those states. Therefore, we adopted the proposal of Müller [8]. Müller specifies that any interaction protocol consists of three general states: *start*, *react* or *complete* depending on the moment in the FSM when the primitive is issued, but we added another the *modify* state to be more specific about a conversation between agents. The set S is represented by $S = \{s1, s2, s3, s4\}$, where $s1$ represents a *start* state, $s2$ represents a *react* state, $s3$ represents a *modify* state, and $s4$ represents a *complete* state.

3.1 Number of Different Interaction Links

To compute pragmatic similarity we need first to calculate the total number of different interaction links and identify the set of pairs of heterogeneous agents that will

interact among them. Considering a set of n agents, the possible number of peer to peer interaction links among them is n^2 . However, as we are evaluating heterogeneity, we need to extract the number of interaction links where agents are equal, which is n . We also considered that an interaction link between agents (a_1, a_2) has the same heterogeneity as an interaction link of agents (a_2, a_1) , thus we reduced the number of different interaction links dividing by 2.

$$\text{Interaction links} = (n^2 - n) / 2 \quad (1)$$

3.2 Algorithm for Computing Pragmatic Similarity

Our algorithm is based on the pragmatic equivalence relation. We adapted this definition in the algorithm, but we are considering only input messages which are syntactically different, and then we are computing similarity among these different messages as follows: if their initial states and their final states are equal, then the input messages are pragmatically similar. To compute pragmatic similarity we implemented an array-based algorithm. For each agent participating in the interaction environment, we need to implement an array, each array with three columns which represent: the initial state, the input primitive and the final state. The algorithm is executed for each different interaction link (a_i, a_j) , where a_i represents the array of agent i . The result of this algorithm is a set of relations, which will help as the basis for a translation approach solution.

```

For each transition function ft of ai
  For each transition function ft of aj
    If (ai[initial-state] is equal to
        aj[initial-state]) and
        (ai[final-state] is equal to
        aj[final-state])
      if (ai[input-primitive]
          is-different-syntactically to
          aj[input primitive])
        ai[input-primitive]
        is-similar-pragmatic to
        aj[input primitive]

```

3.3 Pragmatic Similarity Measure

Another important result is a pragmatic similarity measure, which will help to analyze more precisely the level of similarity of a set of agents. In this section we describe this measure.

The pragmatic similarity measure is a ratio which results from dividing the number of equivalent functions by the total number of transition functions from participating agent's protocols.

1. *Number of transition functions*

The total number of transition functions (*NTF*) is obtained from the sum operation of all sets of transition functions, TFa_n .

$$NTF = TFa_1 + TFa_2 + \dots + TFa_n \tag{2}$$

2. *Number of equivalent functions*

The total number of equivalent functions (*NEF*) results from the sum of the resulting set of the algorithm.

3. *Pragmatic similarity*

The pragmatic similarity results from dividing the number of equivalent functions by *NFT*, which is the ratio that will serve as an indicator for evaluating pragmatic similarity.

$$Pragmatic\ similarity = NEF / NFT \tag{3}$$

4 An Example

In this section we present an example to show the applicability of our approach. Given a MAS integrated with three agents: a_1 , a_2 , and a_3 , each with its own input primitives IP.

$$MAS = \{ a_1, a_2, a_3 \}$$

For each agent there is a set of input primitives which are used as communicative acts to send and receive messages among them. The set of input primitives for each agent are as follows.

$IPa_1 = \{ Initial_Offer, RFQ, Accept, Reject, Offer, Counter_Offer \}$

$IPa_2 = \{ CFP, Propose, Accept, Terminate, Reject, Acknowledge, Modify, Withdraw \}$

$IPa_3 = \{ Requests_Add, Authorize_Add, Require, Demand, Accept, Reject, Unable, Require_for, Insist_for, Demand_for \}$

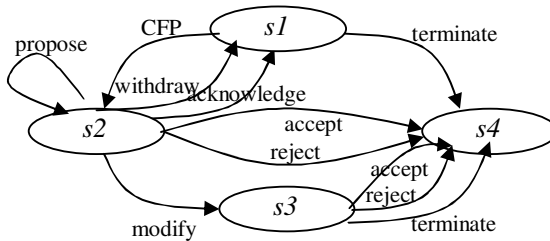


Fig. 1. State transition diagram of agent a_1

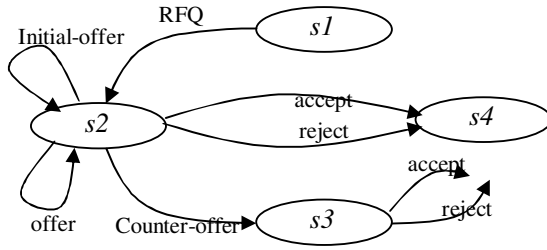


Fig. 2. State transition diagram of agent a_2

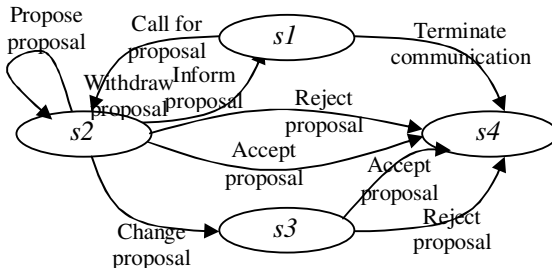


Fig. 3. State transition diagram of agent a_3

Based on the set of input primitives described we manually generate state transition diagrams to compute similarity. In figures 1, 2 and 3 we present the state transition diagram of each agent. In Table 1 we describe the set of transition functions of agents.

Table 1. Transition functions of agents

Transition functions of interaction protocol of agent a_1	Transition functions of interaction protocol of agent a_2	Transition functions of interaction protocol of agent a_3
ft(start, CFP) = react ft(react, Propose) = react ft(react, Acknowledge) = start ft(react, Modify) = modify ft(react, Withdraw) = start ft(react, Reject) = finalize ft(react, Accept) = finalize ft(modify, Reject) = finalize ft(modify, Accept) = finalize ft(react, Terminate) = finalize ft(start, Terminate) = finalize	ft(start, RFQ) = react ft(react, Initial-offer) = react ft(react, Counter-offer) = modify ft(react, Offer) = react ft(modify, Accept) = finalize ft(modify, Reject) = finalize ft(react, Reject) = finalize ft(react, Accept) = finalize	ft(start, Call for proposal) = react ft(react, Propose proposal) = react ft(react, Inform Proposal) = start ft(react, Change Proposal) = modify ft(react, Withdraw Proposal) = react ft(react, Reject Proposal) = finalize ft(react, Accept Proposal) = finalize ft(modify, Reject Proposal) = finalize ft(modify, Accept Proposal) = finalize ft(start, Terminate Communication) = finalize

4.1 Number of Different Interaction Links

We apply Formula 1 to obtain the set of different interaction links for these agents.

$$\begin{aligned} \text{Interaction links} &= (n^2 - n) / 2 \\ \text{Interaction links} &= (3^2 - 3) / 2 = 3 \\ \text{Set of different interaction links} &= \{ (a_1, a_2), (a_1, a_3), (a_2, a_3) \} \end{aligned}$$

4.2 Executing the Algorithm for Computing Pragmatic Similarity

We implemented three arrays with the messages from each agent participating in the interaction environment, each array with three columns which represent: the initial state, the input primitive and the final state. The algorithm is executed for each different interaction link (a_i, a_j) . The result of this algorithm is a set of pragmatic similar relations. Results of this process are shown in Table 2. To define relations we used the form:

$$REL(a_i, P_i, a_j, P_j)$$

where
 a_i is the agent issuer of primitive P_i
 a_j is the agent issuer of primitive P_j

After obtaining the resulting set of similar functions, we have to evaluate it, in order to check inconsistencies. We defined only similar pragmatic relations for primitives that are syntactically different. We did not established differences as relations, because this kind of relations will not support interoperability. However, they are important to measure heterogeneity and to propose another solution based on a learning approach.

Table 2. Resulting set of pragmatic relations

Interaction link (a_1, a_2)	Interaction link (a_1, a_3)	Interaction link (a_2, a_3)
REL(a_1 , CFP, a_2 , RFQ)	REL(a_1 , CFP, a_3 , Call for proposals)	REL(a_2 , RFQ, a_3 , Call for proposals)
REL(a_1 , Propose, a_2 , Initial_Offer)	REL(a_1 , Propose, a_3 , Propose proposal)	REL(a_2 , Offer, a_3 , Propose proposal)
REL(a_1 , Modify, a_2 , Counter_offer)	REL(a_1 , Modify, a_3 , Change proposal)	REL(a_2 , Counter_offer, a_3 , Change proposal)
REL(a_1 , Propose, a_2 , Offer)	REL(a_1 , Withdraw, a_3 , Withdraw proposal)	REL(a_2 , Accept, a_3 , Accept proposal)
REL(a_1 , Terminate, a_2 , Reject)	REL(a_1 , Acknowledge, a_3 , Inform proposal)	REL(a_2 , Reject, a_3 , Reject proposal)
REL(a_1 , Terminate, a_2 , Accept)	REL(a_1 , Accept, a_3 , Accept proposal)	REL(a_2 , Initial_Offer, a_3 , Propose proposal)
	REL(a_1 , Reject, a_3 , Reject proposal)	
	REL(a_1 , Terminate, a_3 , Terminate communication)	

4.3 Pragmatic Similarity Measure

Another important result is the pragmatic similarity measure, which will help to analyze more precisely the level of similarity between these agents. The resulting pragmatic similarity ratios for the three interaction links is shown in Table 3.

Table 3. Pragmatic similarity ratios

IL	NEF	NTF	Pragmatic Similarity
(a_1, a_2)	6	19	.31
(a_1, a_3)	8	21	.38
(a_2, a_3)	6	18	.33

Therefore a ratio of 1 indicates a fully pragmatic interoperability between agents, while a ratio of 0 indicates impossibility of interoperation between agents.

5 Conclusions

In this paper we have described how to represent and compute pragmatic similarity, to generate a set of pragmatic relations among agent protocols and to measure the similarity in MAS. FSM is a good formalism for representing interaction scenarios between agents, in particular they are suitable to compare interaction protocols and identify the state of a conversation in a MAS environment. Our approach is practical, because it is based on FSM definitions; as a result we have implemented an algorithm based on the pragmatic equivalence relation definition.

In contrast to the semantic approach, which works with data sources, we propose a different approach for comparing software pragmatics based on their message passing logics. The result of this analysis helps the developer to measure pragmatic similarity, in order to design and implement a better solution.

The set of defined relations were implemented in an ontology-based translator which showed better results in interaction environments when the translator was invoked. For a deeper description of this execution environment, please refer to [9].

This is a promising research area, because nowadays there is a tremendous amount of legacy software which in turn will require to be incorporated independently of the inherent heterogeneity inside their logics or protocols.

Our pragmatic approach can be applied in other application areas such as Web service discovery, process engineering, program comparison or application integration, which have in common that they provide functionality information similar to protocols in interaction scenarios.

References

1. Petri, C.A.: Kommunikation mit Automaten. Ph. D. Thesis. University of Bonn (1962)
2. Scott Cost, R., Chen, Y., Finin, T., Labrou, Y., Peng, Y.: Modeling agent conversations with colored petri nets. In: Working notes of the Workshop on Specifying and Implementing Conversation Policies, Seattle, Washington, pp. 59–66 (1999)
3. Milner, R.: Communicating and Mobile Systems: The Pi-Calculus. Cambridge Univ. Press, Cambridge (1999)
4. Odell, J., Van Dyke Parunak, H., Buer, B.: Representing agent interaction protocols in UML. In: Ciancarini, P., Wooldridge, M.J. (eds.) AOSE 2000. LNCS, vol. 1957, pp. 121–140. Springer, Heidelberg (2001)
5. BPEL, Business process Execution Language for Web Services. Version 1.1 (May 2003)
6. Ankolekar, A., Burstein, M., Hobbs, J.R., Lassila, O., Martin, D., McDermott, D., McIlraith, S., Narayanan, S., Paloucci, M., Payne, T., Sycara, K.: DAML-S: Web service description for the semantic Web. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, Springer, Heidelberg (2002)
7. Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley, Reading (2001)
8. Müller, H.J.: Negotiation Principles. In: O'Hare, G.M.P., Jennings, N.R. (eds.) Foundations of Distributed Artificial Intelligence, John Wiley & Sons, New York (1996)
9. Bravo, M., Pérez, J., Velázquez, J., Sosa, V., Montes, A., López, M.: Design of a Shared Ontology Used for Translating Negotiation Primitives. International Journal of Web and Grid Services 2(3), 237–259 (2006)

On the Relevance of Organizational Structures for a Technology of Agreement

Holger Billhardt, Roberto Centeno, Alberto Fernández,
Ramón Hermoso, Rubén Ortiz, Sascha Ossowski, and Matteo Vasirani

Centre for Intelligent Information Technologies (CETINIA),
Universidad Rey Juan Carlos,
Calle Tulipán s/n,
28933 Móstoles (Madrid), Spain

Abstract. This paper provides a brief overview of the field of coordination in multi-agent systems, and outlines its relation to current efforts working towards a paradigm for smart, next-generation distributed systems, where coordination is based on the concept of agreement between computational entities. To illustrate the types of mechanisms that we envision to be part of a “technology of agreement”, we provide two examples of how techniques from the field of organisations can be used to foster coordination and agreement in open multi-agent systems.

1 Introduction

An increasing number of transactions and interactions at business level, but also at leisure level, are nowadays mediated by computers and computer networks. An appealing way to model and design such applications is by purposefully combining components to which more and more complex tasks can be delegated. These components need to show an adequate level of intelligence, should be capable of sophisticated ways of interacting, and are usually massively distributed, sometimes embedded in all sort of appliances and sensors. In order to allow for an efficient design and implementation of systems of these characteristics, it is necessary to effectively enable, structure, and regulate their communications in different contexts.

Such an enterprise raises a number of technological challenges. Firstly, the open distributed nature of such systems adds to the *heterogeneity* of its components. The system structure may evolve at runtime, as new nodes may appear or disappear at will. There is also a need for on-the-fly alignment of certain concepts that interactions relate to, as the basic ontological conventions in such systems will be very limited. The *dynamicity* of the environment calls for a continuous *adaptation* of the structures that regulate the components’ interactions, so as to achieve and sustain desired functional properties. But also non-functional issues related to *scalability*, *security*, and *usability* need to be taken into account. When designing mechanisms that address these challenges, the notion of *autonomy* becomes central: components may show complex patterns of activity aligned with the different goals of their designers, while it is usually impossible to directly influence their behaviour from the outside.

Coordination in multi-agent system (MAS) aims at harmonising the interactions of multiple autonomous components or agents. Therefore, it appears promising to review different conceptual frameworks for MAS coordination, and to analyse the potential and limitations of the work done in that field with regard to some of the aforementioned challenges.

This paper is organised as follows. Section 2 provides a brief overview of coordination in MAS, identifies the notion of *agreement* as a centrepiece of an integrated approach to coordination in open distributed systems, and outlines some research topics related to the vision of a technology of agreement. Section 3 provides examples of how organisational structures can be used to instil coordination and agreement in open multi-agent systems, in the realm of matchmaking and trust mechanisms. Some conclusions are drawn in Section 4.

2 Coordination and Agreement in Multi-agent Systems

Maybe the most widely accepted conceptualisation of coordination in the MAS field originates from Organisational Science. It defines coordination the *management of dependencies* between organisational activities [20]. In a multi-agent setting, the subjects whose activities need to be coordinated are the agents, while the entities between which dependencies are usually goals, actions or plans. Depending on the characteristics of the MAS environment, a taxonomy of dependencies can be established, and a set of potential coordination actions assigned to each of them (e.g. [36], [23]). Within this model, the *process* of coordination is to accomplish two major tasks: first, a *detection* of dependencies needs to be performed, and second, a *decision* respecting which coordination action to apply must be taken. A coordination *mechanism* shapes the way that agents perform these tasks [21].

From a *macro-level* (MAS-centric) perspective, the outcome of coordination can be conceived a “global” plan (or decision, action etc.). This may be a “joint plan” [28] if the agents reach an explicit agreement on it during the coordination process, or just the sum of the agents’ individual plans (or decisions, actions etc. – sometimes called “multi-plan” [24]) as perceived by an external observer. Roughly speaking, the quality of the outcome of coordination at the macro-level can be evaluated with respect to the agents’ joint goals or the desired functionality of the MAS as a whole. If no such notion can be ascribed to the MAS, other, more basic features can be used instead. A good result of coordination, for instance, often relates to “efficiency”, which frequently comes down to the notion of Pareto-optimality. The amount of resources necessary for coordination (e.g. the number of messages necessary) is also sometimes used as a measure of efficiency.

The dependency model of coordination appears to be particularly adequate for *representing* relevant features of coordination problems in MAS. Frameworks based on this model have been used to capture coordination requirements in a variety of interesting MAS domains (e.g. [8]). Still, dependency detection may become a rather knowledge intensive task, which is further complicated by incomplete and potentially inconsistent local views of the agents. From a design perspective, coordination is probably best conceived as the effort of *governing the space of interaction* [5] of a MAS, as the basic challenge amounts to how to make agents converge on interaction

patterns that adequately (i.e. instrumentally with respect to desired MAS features) solve the dependency detection and decision tasks. A variety of approaches that tackle this problem can be found in the literature, shaping the interaction space either directly, by making assumptions on agent behaviours and/or knowledge, or indirectly, by modifying the agent's environment [30] (e.g. the MAS infrastructure [22], or the institutional context [10]). The applicability of these mechanisms depends largely on the number and type of assumptions that one may make regarding the possibility of manipulating agent programs, agent populations, or the agents' environment. This, in turn, is dependent on the characteristics of the coordination problem at hand.

From the point of view of an individual agent, the problem of coordination boils down to finding the sequence of actions that, given the regulations within the system (or, if this possible in a certain environment, the expected cost of transgressing them), best achieves its goals. In practice, this implies a series of non-trivial problems. Models of coalition formation determine when and with whom to form a team for the achievement of some common (sub-) goal, and how to distribute the benefits of synergies that arise from this cooperation [32]. Distributed planning approaches [9] determine how to (re-)distribute tasks among team members and how to integrate results. From an individual agent's perspective, the level of trustworthiness of others is central to almost every stage of these processes, so as to determine whether other agents are likely to honour the commitments that have been generated [33].

Several quite different approaches and mechanisms coexist under the "umbrella" of the term coordination in MAS [25]. Not all of them are relevant to the challenges for the design of open distributed systems outlined in the introduction. For instance, the whole set of *coupled* coordination mechanisms [35] are effectively useless for the purpose of this paper, as they require having a direct influence on the agent programs. On the other hand, the problem of semantic interoperability is usually outside the scope of MAS coordination models and languages.

The notion of *agreement* among computational agents appears to be better suited as the fundamental notion for the proposal outlined in this paper. Following a recent research effort in the field of "Agreement Technologies" [1], the process of agreement-based coordination can be conceived based on two main elements:

- (1) a normative context, that determines the rules of the game, i.e. interaction patterns and additional restrictions on agent behaviour; and
- (2) a call-by-agreement interaction method, where an agreement for action between the agents that respects the normative context is established first; then actual enactment of the action is requested.

Methods and mechanisms from the fields of *semantic alignment*, *norms*, *organization*, *argumentation and negotiation*, as well as *trust and reputation* are envisioned be part of a "sandbox" to build software systems based on a technology of agreement [1].

Semantic technologies should constitute a centrepiece of such an enterprise as semantic problems pervade all the others. Solutions to semantic mismatches and alignment of ontologies [4] are needed to have a common understanding of norms or of deals, just to put two examples. As we will illustrate in the next section, the use of semantics-based approaches to service discovery and composition will allow exploring the space of possible interactions and, consequently, shaping the set of possible agreements [12].

At system-level, *norms* are needed to determine constraints that the agreements, and the processes to reach them, have to satisfy. Reasoning about a system's norms is necessary at design-time to assure that the system has adequate properties, but it may also be necessary at run-time, as complex systems usually need dynamic regulations [14]. *Organisational* structures further restrict the way agreements are reached by fixing the social structure of the agents: the capabilities of their roles and the relationships among them (e.g. power, authority) [3].

Moving further towards the agent-level, *negotiation* methods are essential to make agents reach agreements that respect the constraints imposed by norms and organisations. These methods need to be complemented by an argumentation-based approach: by exchanging arguments, the agents' mental states may evolve and, consequently, the status of offers may change [2] [6]. Finally, agents will need to use *trust* mechanisms that summarise the history of agreements and subsequent agreement executions in order to build long-term relationships between the agents [34].

Of course, these methods should not be seen in isolation, as they may well benefit from each other. For instance, in certain situations trust mechanisms may take advantage of the roles structures included in an organisational model, so as to improve their performance when only limited information about previous interactions is available.

3 Organizational Structures and Agreement

This section intends to illustrate the types of mechanisms that we envision being part of the agreement technology "sandbox" mentioned previously. In particular, we will provide examples of how organisational structures can be used to foster coordination and agreement in open MAS.

Organisational models underlying approaches such as Agent-Group-Role [11], MOISE [16], or RICA [31] provide a rich set of concepts to specify and structure mechanisms that govern agent interactions through the corresponding infrastructures or middleware. A key notion in most organisational models is the concept of role. Roles can often be organised in a taxonomy, which can be modelled as a pair $\langle R, \leq \rangle$ where R is the set of concepts representing roles and \leq is a partial order among R .

In section 3.1, we show how role taxonomies can be used to locate suitable interactions partners, by providing additional information regarding the usability of services in a certain interaction context. Section 3.2 outlines how such taxonomies can be used for the bootstrapping of reputation mechanisms, when only limited information about past interactions is available in the system.

3.1 Organisational Structures and Matchmaking Mechanisms

Our first example refers to service-oriented MAS where the capabilities of agents are modelled in the shape of services which, in turn, are described by some standard service description language. In the following we present our approach to enriching service descriptions with organisational information. For this purpose, we first introduce simple languages for representing role-based service advertisements and service requests.

A service advertisement S is a set of pairs so that

$$S \subseteq \left\{ \langle r, \rho \rangle \mid r \in R, \rho = \bigvee_{i=1}^n \bigwedge_{j=1}^m r_{ij}, r_{ij} \in R \right\}$$

In this definition, r is the role played by the provider in the interaction, and ρ is a set of roles that must be played by the requester agent for the correct accomplishment of the service, given by a formula in disjunctive normal form (DNF).

A service request Q is a set of pairs so that

$$Q \subseteq \langle \rho, C \rangle, \rho = \bigvee_{i=1}^n \bigwedge_{j=1}^m r_{ij}, r_{ij} \in R, C \subseteq R$$

Again, ρ is a DNF role expression (usually atomic) specifying the searched provider roles, and C is a set of roles that define the *capabilities* of the requester (the roles it is able to play).

Although organisational information is not a first-class citizen in service description languages such as OWL-S¹ or WSMO², it is not difficult to incorporate it into them. In OWL-S, for instance, we propose to include the role description as an additional parameter, called *Service_Roles*, in the case of service descriptions (r and ρ are mapped to *providerRole* and *dependingRoles* tags, respectively), and *Query_Roles* for service requests (ρ and C are mapped to *SearchedProviderRoles* and *CapabilityRoles*) [12].

In many multi-agent settings, this kind of organisational information can be used to complement standard I/O based matchmaking in order to improve its performance. We set out from the following requirements to define a semantic match function between two roles:

1. It must return a real number in the range [0..1] (degree of match or *dom*), with a higher value the more similar the concepts are (1 if $r_1=r_2$).
2. It must consider the distance between both concepts (roles) in the ontology: the greater the distance, the less similar are the concepts (decreasing function).
3. The change of *dom* per unit must decrease inversely with the distance (e.g., the step from 1 to 2 is more relevant than 5 to 6).
4. The $dom(r_1, r_2)$ must be independent of the height of the taxonomy and its location within it.
5. The logical relation between the two roles (i.e. the subsumption relation) must be taken care of. This is the most important criterion to take into account.

Requirement 2 is addressed by using the measure proposed by Rada [27], consisting of the number of edges in the shortest path between two concepts in the taxonomy:

$$dist(c_1, c_2) = depth(c_1) + depth(c_2) - 2 \times depth(lcs^3(c_1, c_2))$$

Requirement 3 imposes a non-linear decreasing function. We use a typical exponential function here, $e^{-dist(r_1, r_2)}$, as it maintains its range in [0..1], is monotonically

¹ <http://www.daml.org/services/owl-s/>

² <http://www.wsmo.org/>

³ Least common subsumer

decreasing, is 1 when $r_1=r_2$ (requirement 1), and it does neither depend on the height of the taxonomy nor on the global height of the roles (requirement 4).

In order to comply with requirement 5, we differentiate among the four levels of match proposed by Paolucci et al. [26] (advertisement A and request Q):

- *exact*: if $r_A = r_Q$
- *plug-in*: if r_A *subsumes* r_Q
- *subsumes*: if r_Q *subsumes* r_A
- *fail*: otherwise

We take the final value, representing the degree of match, equal to 1 in case of an *exact* match, it varies between 1 and 0.5 in case of a *plug-in* match, stays between 0.5 and 0 in case of a *subsumes* match, and it is equal to 0 in case of a *fail*. So we only have to scale the value [0..1] to the ranges [0..0.5] and [0.5..1].

Based on these considerations, we define the degree of matching *dom* between two roles R_A and R_Q as

$$dom(R_A, R_Q) = \begin{cases} 1 & \text{if } R_A = R_Q \\ \frac{1}{2} + \frac{1}{2 \cdot e^{\|R_A, R_Q\|}} & \text{if } R_A \text{ is subclass of } R_Q \\ \frac{1}{2} \cdot e^{\|R_A, R_Q\|} & \text{if } R_Q \text{ is subclass of } R_A \\ 0 & \text{otherwise} \end{cases}$$

where $\|R_A, R_Q\|$ is the distance between R_A and R_Q ($dist(R_A, R_Q)$) in the role taxonomy (if there is a subsumption relation between them). By construction, this equation fits the requirements.

The *semantic match* between a service advertisement S and a query Q (service request) is done by searching the role in S that best matches the one in Q . The degree of match between a role in the request and a service advertisement, given the set of capabilities of the requester, is done by comparing the searched role with every other given role and returns the maximum degree of match. For each role in the advertisement, the match between the provider roles is made, as well as the match between the depending roles and the capabilities of the requester.

The minimum of both values is considered the degree of match. In case of logical expressions, the minimum is used as combination function for the values in a conjunction and the maximum for disjunctions (which always keep the value resulting of the combination within the range [0,1]). Details of the algorithm used to determine the degree of match between a service request and a service advertisement are described in [12].

Our approach is intended to be complementary to other general-purpose matchmakers. We have performed experiments combining an implementation of the semantic match between services (ROWLS) with OWLS-MX [19], one of the leading hybrid matchmakers available to-date. Comparing a combination of ROWLS and OWL-MX to a standalone use of the latter, we have found an improvement to both effectiveness and efficiency based on our test collection [12].

3.2 Organisational Structures and Trust Mechanisms

The second example shows how an agent can use knowledge about the organisational structure to infer confidence in a situation when no previous experience about a specific interaction is available. Similar to other approaches [17][29], we set out from a trust model based on the idea of *confidence* and *reputation*. Both ratings evaluate the trustworthiness of other agents in a particular situation (e.g., playing a particular role in a particular interaction). *Confidence* is a local measure that is only based on an agent's own experiences, while *reputation* is an aggregated value an agent gathers by asking its acquaintances about their opinion regarding the trustworthiness of another agent. Thus, reputation can be considered as an external *social* measure. We define *trust* as a rating resulting from combining *confidence* and *reputation* values.

A typical scenario for the use of a trust model is the following. An agent A wants to evaluate the trustworthiness of some other agent B – playing the role R – in the interaction I . This trustworthiness is denoted as $t_{A \rightarrow \langle B,R,I \rangle} \in [0..1]$, measuring the trust of A in B (playing role R) being a “good” counterpart in the interaction I . When evaluating the trustworthiness of a potential counterpart, an agent can combine its local information (confidence) with the information obtained from other agents regarding the same counterpart (reputation).

Confidence, $c_{A \rightarrow \langle B,R,I \rangle}$, is collected from A 's past interactions with agent B playing role R and performing interactions of type I . We call *Local Interaction Table* (LIT) an agent's data structure storing confidence values for past interactions with any counterpart the agent has interacted with. Each entry corresponds to a *situation*: an *agent* playing a specific *role* in a particular *interaction*. LIT_A denotes agent A 's LIT. An example is shown in Table 1. Each entry in a LIT consists of: (i) the Agent/Role/Interaction identifier $\langle X,Y,Z \rangle$, (ii) the confidence value for the issue ($c_{A \rightarrow \langle X,Y,Z \rangle}$), and (iii) a reliability value ($r_{A \rightarrow \langle X,Y,Z \rangle}$). The confidence value is obtained from some function that evaluates past experiences on the same situation. We suppose $c_{A \rightarrow \langle X,Y,Z \rangle} \in [0..1]$ where higher values represent higher confidence.

Table 1. An agent's local interaction table (LIT_A)

$\langle X,Y,Z \rangle$	$c_{A \rightarrow \langle X,Y,Z \rangle}$	$r_{A \rightarrow \langle X,Y,Z \rangle}$
$\langle a_9, r_2, i_3 \rangle$	0.2	0.75
$\langle a_2, r_7, i_1 \rangle$	0.7	0.3
\vdots	\vdots	\vdots
$\langle a_9, r_2, i_5 \rangle$	0.3	0.5

Each direct experience of an agent regarding a situation $\langle X,Y,Z \rangle$ changes its confidence value $c_{A \rightarrow \langle X,Y,Z \rangle}$. In this sense, we suppose that the agents have some mechanism to evaluate the behaviour of other agents that they interact with. Let $g_{\langle X,Y,Z \rangle} \in [0..1]$ denote the evaluation value an agent A calculates for a particular

experience with the agent X playing role Y in the interaction of type Z . We use the following formula to update confidence:

$$c_{A \rightarrow \langle X, Y, Z \rangle} = \varepsilon \cdot c'_{A \rightarrow \langle X, Y, Z \rangle} + (1 - \varepsilon) \cdot g_{\langle X, Y, Z \rangle}$$

where $c'_{A \rightarrow \langle X, Y, Z \rangle}$ is the confidence value in A 's LIT before the interaction is performed and $\varepsilon \in [0..1]$ is a parameter specifying the importance given to A 's past confidence value. In general, the aggregated confidence value from past experiences will be more relevant than the evaluations of the most recent interactions.

Reliability ($r_{A \rightarrow \langle X, Y, Z \rangle}$) measures how certain an agent is about its own confidence in a situation. We suppose $r_{A \rightarrow \langle X, Y, Z \rangle} \in [0..1]$. Furthermore, we assume that $r_{A \rightarrow \langle X, Y, Z \rangle} = 0$ for any tuple $\langle X, Y, Z \rangle$ not belonging to LIT_A . We calculate reliability by using the approach proposed by Huynh, Jennings and Shadbolt [18], taking into account the number of interactions a confidence value is based on and the variability of the individual values across past experiences.

An agent may build trust directly from its confidence value or it may combine confidence with reputation. Reputation is particularly useful when an agent has no experience or if the reliability value for the confidence is not high. Social reputation may be obtained by asking other agents about their opinion on a situation. Agents that have been requested for their opinion will return the corresponding confidence and reliability ratings from their LIT. The requester might then be able to build trust by calculating a weighted mean over its own confidence value and the confidence values received from others, as it is represented in the following equation:

$$t_{A \rightarrow \langle B, R, I \rangle} = \begin{cases} c_{A \rightarrow \langle B, R, I \rangle} & \text{if } r_{A \rightarrow \langle B, R, I \rangle} > \theta \\ \frac{\sum_{x \in AA \cup \{A\}} c_{x \rightarrow \langle B, R, I \rangle} \cdot \omega_{x \rightarrow \langle B, R, I \rangle}}{\sum_{x \in AA \cup \{A\}} \omega_{x \rightarrow \langle B, R, I \rangle}} & \text{otherwise} \end{cases}$$

$\theta \in [0..1]$ is a threshold on the reliability of confidence. If the reliability is above θ then an agent's own confidence in a situation is used as the trust value. Otherwise trust is built by combining confidence and reputation. AA is a set of acquaintances an agent asks about their opinion regarding the situation $\langle B, R, I \rangle$. For instance, in some scenarios it may be useful to ask other agents that play the same role as A , since they may have similar interests and goals.

The weights $\omega_{x \rightarrow \langle B, R, I \rangle}$ given to the gathered confidence values are composed of the corresponding reliability values and a constant factor α that specifies the importance given to A 's own confidence in the issue:

$$\omega_{x \rightarrow \langle B, R, I \rangle} = \begin{cases} r_{x \rightarrow \langle B, R, I \rangle} \cdot \alpha & \text{if } x = A \\ r_{x \rightarrow \langle B, R, I \rangle} \cdot (1 - \alpha) & \text{otherwise} \end{cases}$$

Basic trust models as the one outlined before run into problems when no interactions of a specific type have been performed before and, in addition, social reputation

is not available or not reliable. In such a situation, information of the organisational structure can be used to determine an approximate degree of trust.

In particular, one approach consists of using the agent/role confidence $c_{A \rightarrow \langle B, R, \dots \rangle}$ (or the agent confidence $c_{A \rightarrow \langle B, \dots \rangle}$) as an estimation for $c_{A \rightarrow \langle B, R, I \rangle}$ if agent A has no reliable experience about situation $\langle B, R, I \rangle$. This approach relies on the hypothesis that, in general, *agents behave in a similar way in all interactions related to the same role*. We argue that, exploiting this idea, the more similar I' and I are, the more similar the values $c_{A \rightarrow \langle B, R, I' \rangle}$ and $c_{A \rightarrow \langle B, R, I \rangle}$ will be. The same applies to roles.

Taking this assumption further, confidence ratings for similar agent/role/interaction tuples can be accumulated to provide evidence for the trustworthiness of the situation $\langle B, R, I \rangle$. Based on this idea, we propose to build trust by taking into account all the past experiences an agent has, focusing on their degree of similarity between organisational concepts, with the situation $\langle B, R, I \rangle$. In particular, we calculate trust as a weighted mean over all the confidence values an agent has accumulated in its LIT. This is shown in the following equation:

$$t_{A \rightarrow \langle B, R, I \rangle} = \frac{\sum_{\langle X, Y, Z \rangle \in \text{LIT}_A} c_{A \rightarrow \langle X, Y, Z \rangle} \cdot \omega_{A \rightarrow \langle X, Y, Z \rangle}}{\sum_{\langle X, Y, Z \rangle} \omega_{A \rightarrow \langle X, Y, Z \rangle}}$$

$\omega_{A \rightarrow \langle X, Y, Z \rangle}$ is the weight given to agent A 's confidence on situation $\langle X, Y, Z \rangle$. The weights combine the confidence reliability with the similarity of the situation $\langle X, Y, Z \rangle$ to the target issue $\langle B, R, I \rangle$ in the following way:

$$\omega_{A \rightarrow \langle X, Y, Z \rangle} = r_{A \rightarrow \langle X, Y, Z \rangle} \cdot \text{sim}(\langle X, Y, Z \rangle, \langle B, R, I \rangle)$$

The similarity function $\text{sim}(\langle X, Y, Z \rangle, \langle B, R, I \rangle)$ is computed as the weighted sum of the similarities of the individual elements (agent, role and interaction) as it is shown in the following equation:

$$\text{sim}(\langle X, Y, Z \rangle, \langle B, R, I \rangle) = \begin{cases} \beta \cdot \text{sim}_R(R, Y) + \gamma \cdot \text{sim}_I(I, Z) & \text{if } X = B \\ 0 & \text{otherwise} \end{cases}$$

where $\text{sim}_R(R, Y), \text{sim}_I(I, Z) \in [0..1]$ measures the similarity between roles and interactions, respectively, and β and γ with $\beta + \gamma = 1$, are parameters specifying the sensibility regarding the individual similarities.

Role similarities can be inferred from role taxonomies contained in an organisational model. In particular, $\text{sim}_R(R, R')$ can rely on a *distance function*, similar to the one presented in the previous subsection, that estimates the similarity between two roles on the basis of their proximity in the taxonomy. The same holds for $\text{sim}_I(I, I')$ when an interaction taxonomy is available in an organisational model [15].

Especially if an agent has no reliable experience about a particular agent/role/interaction situation, our organisation-based approach can be used to estimate trust without the necessity to rely on the opinions of other agents. So, role and interaction

taxonomies can help making agents that use trust mechanisms less vulnerable to dishonest counterparts, as there is less need to rely on third-party information.

4 Discussion

This paper has presented an overview of different approaches to coordination in the MAS field. It has been argued that the notion of agreement is essential to instil coordination in open distributed systems. Some existing technologies from the field of MAS coordination can be applied to this respect, but others – and in particular semantic technologies – need to be added. To illustrate the types of mechanisms that we envision to be part of a “technology of agreement”, we have provided two examples of how techniques from the field of organisations can be used to foster coordination and agreement in open MAS.

We have shown how organisational structures can be used to complement traditional matchmaking mechanisms so as to enhance their performance. We are currently evaluating as to how far more fine-grained quantitative matching techniques can be applied to this respect [13]. Furthermore, we have argued that organisational structures can be used to improve reputation mechanisms in situations where only a limited amount of information regarding previous interactions is available. Current work focuses on how, in turn, the history of interactions can be used to evolve organisational structures [15].

Several research efforts are currently ongoing that may contribute to the development of a “technology of agreement” in one or another way. The attempt to harmonise these efforts, which is currently being carried out at European level, promotes the emergence of a new paradigm for next generation distributed systems based on the notion of *agreement* between computational agents [7].

Acknowledgements

Some ideas reported in this paper draw upon joint work with our partners in the framework of a Spanish national project on “Agreement Technology”. This work was partially supported by the Autonomous Region of Madrid, grant URJC-CM-2006-CET-0300, and by the Spanish Ministry of Science and Innovation, grants TIN2006-14630-C03-02 and CSD2007-00022 (CONSOLIDER-INGENIO 2010).

References

- [1] Agreement Technologies project homepage, <http://www.agreement-technologies.org/>
- [2] Amgoud, L., Dimopolous, Y., Moraitis, P.: A unified and general framework for argumentation-based negotiation. In: Proc. 6th Int. Joint Conference on Autonomous Agents and Multi-Agents Systems (AAMAS 2007). IFAAMAS, pp. 963–970 (2007)
- [3] Argente, E., Julian, V., Botti, V.: Multi-Agent System Development based on Organizations. Elec. Notes in Theoretical Computer Science 150(3), 55–71 (2006)

- [4] Atienza, M., Schorlemmer, M.: I-SSA - Interaction-situated Semantic Alignment. In: Proc Int. Conf. on Cooperative Information Systems (CoopIS 2008) (to appear, 2008)
- [5] Busi, N., Ciancarini, P., Gorrieri, R., Zavattaro, G.: Coordination Models - A Guided Tour. In: Omicini, et al. (eds.) *Coordination of Internet Agents: Models, Technologies, and Applications*, pp. 6–24. Springer, Heidelberg (2001)
- [6] Caminada, M., Amgoud, L.: On the evaluation of argumentation formalisms. *Artificial Intelligence Journal* 171(5-6), 286–310 (2007)
- [7] COST Act. IC0801,
http://www.cost.esf.org/index.php?id=110&action_number=IC0801
- [8] Decker, K.: TAEMS: A Framework for Environment Centered Analysis and Design of Coordination Mechanisms. In: O'Hare, Jennings (eds.) *Foundations of Distributed Artificial Intelligence*, John Wiley and Sons, Chichester (1996)
- [9] Durfee, E.: Distributed Problem Solving and Planning. In: Luck, M., Mařík, V., Štěpánková, O., Trapp, R. (eds.) *ACAI 2001 and EASSS 2001*. LNCS (LNAI), vol. 2086, pp. 118–149. Springer, Heidelberg (2001)
- [10] Esteva, M., Rosell, B., Rodríguez-Aguilar, J.A., Arcos, J.L.: AMELI - An agent-based middleware for electronic institutions. In: Proc. Int. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), pp. 236–243. ACM Press, New York (2004)
- [11] Ferber, J., Gutknecht, O., Fabien, M.: From Agents to Organizations - An Organizational View of Multi-agent Systems. In: Giorgini, P., Müller, J.P., Odell, J.J. (eds.) *AOSE 2003*. LNCS, vol. 2935, pp. 214–230. Springer, Heidelberg (2004)
- [12] Fernández, A., Ossowski, S.: Exploiting Organisational Information for Service Coordination in Multiagent Systems. In: Proc. of the Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008). IFAAMAS, pp. 257–264 (2008)
- [13] Fernández, A., Polleres, A., Ossowski, S.: Towards Fine-grained Service Matchmaking by Using Concept Similarity. In: Noia, D., et al. (eds.) *Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web (SMR2) CEUR Workshop Proceedings*, vol. 243, pp. 31–34 (2007)
- [14] Gaertner, D., García-Camino, A., Noriega, P., Rodríguez-Aguilar, J.A., Vasconcelos, W.: Distributed norm management in regulated multiagent systems. In: Proc. Int. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007). IFAAMAS, pp. 624–631 (2007)
- [15] Hermoso, R., Centeno, R., Billhardt, H., Ossowski, S.: Extending Virtual Organizations to improve trust mechanisms (Short Paper). In: Proc. Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008). IFAAMAS, pp. 1489–1492 (2008)
- [16] Hubner, J., Sichman, J., Boissier, O.: Developing organised multiagent systems using the MOISE+ model: programming issues at the system and agent levels. *Int. Journal of Agent-Oriented Software Engineering* 1(3/4), 370–395 (2006)
- [17] Huynh, T.D., Jennings, N.R., Shadbolt, N.: FIRE: An integrated trust and reputation model for open multi-agent systems. In: Proceedings of the 16th European Conference on Artificial Intelligence (ECAI), pp. 18–22 (2004)
- [18] Huynh, T.D., Jennings, N.R., Shadbolt, N.R.: Developing an integrated trust and reputation model for open multi-agent systems. In: Proceedings of 7th International Workshop on Trust in Agent Societies (AAMAS), pp. 65–74 (2004)
- [19] Klusch, M., Fries, B., Sycara, K.: Automated Semantic Web Service Discovery with OWLS-MX. In: Proc. Int. Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS 2006), pp. 915–922. ACM Press, New York (2006)

- [20] Malone, T., Crowston, K.: The Interdisciplinary Study of Co-ordination. *Computing Surveys* 26(1), 87–119 (1994)
- [21] Omicini, A., Ossowski, S.: Objective versus Subjective Coordination in the Engineering of Agent Systems. In: Klusch, et al. (eds.) *Intelligent Information Agents – The European AgentLink Perspective*, pp. 179–202. Springer, Heidelberg (2003)
- [22] Omicini, A., Ossowski, S., Ricci, A.: Coordination Infrastructures in the Engineering of Multiagent Systems. In: Bergenti, Gleizes, Zambonelli (eds.) *Methodologies and software engineering for agent systems – The Agent-Oriented Software Engineering Handbook*, pp. 273–296. Kluwer, Dordrecht (2004)
- [23] Ossowski, S.: Co-ordination in Artificial Agent Societies. LNCS (LNAI), vol. 1535. Springer, Heidelberg (1998)
- [24] Ossowski, S.: Constraint Based Coordination of Autonomous Agents. *Electronic Notes in Theoretical Computer Science*, vol. 48, pp. 211–226. Elsevier, Amsterdam (2001)
- [25] Ossowski, S., Menezes, R.: On Coordination and its Significance to Distributed and Multi-Agent Systems. *Journal of Concurrency and Computation - Practice and Experience* 18(4), 359–370 (2006)
- [26] Paolucci, M., Kawamura, T., Payne, T., Sycara, K.: Semantic matching of web services capabilities. In: *Proceedings of the First International Semantic Web Conference on The Semantic Web*, pp. 333–334. Springer, Heidelberg (2002)
- [27] Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. *IEEE Trans. on Systems, Man and Cybernetics* 19(1), 17–30 (1989)
- [28] Rosenschein, J., Zlotkin, G.: Designing Conventions for Automated Negotiation. *AI Magazine* 15(3), 29–46 (1995)
- [29] Sabater, J., Sierra, C.: REGRET: a reputation model for gregarious societies. In: *Proceedings of the Fifth International Conference on Autonomous Agents (AGENTS)*, pp. 194–195. ACM Press, New York (2001)
- [30] Schumacher, M., Ossowski, S.: The governing environment. In: Weyns, D., Van Dyke Parunak, H., Michel, F. (eds.) *E4MAS 2005*. LNCS (LNAI), vol. 3830, pp. 88–104. Springer, Heidelberg (2006)
- [31] Serrano, J.M., Ossowski, S.: On the Impact of Agent Communication Languages on the Implementation of Agent Systems. In: Klusch, M., Omicini, A., Ossowski, S., Laamanen, H. (eds.) *CIA 2003*. LNCS (LNAI), vol. 2782, pp. 92–106. Springer, Heidelberg (2003)
- [32] Shehory, O., Sycara, K., y Somesh, J.: Multi-agent Coordination through Coalition Formation. In: Rao, A., Singh, M.P., Wooldridge, M.J. (eds.) *ATAL 1997*. LNCS, vol. 1365, pp. 143–154. Springer, Heidelberg (1998)
- [33] Sabater, J., Sierra, C.: Review on Computational Trust and Reputation Models. *Artificial Intelligence Review* 24(1), 33–60 (2005)
- [34] Sierra, C., Debenham, J.: Information-Based Agency. In: *Proc Int. Joint Conference on AI (IJCAI 2007)*, pp. 1513–1518. AAAI Press, Menlo Park (2007)
- [35] Tolksdorf, R.: Models of Coordination. In: Omicini, Zambonelli, Tolksdorf (eds.) *Engineering Societies in an Agent World*, Springer, Heidelberg (2000)
- [36] Von Martial, F.: Coordinating Plans of Autonomous Agents. LNCS, vol. 610. Springer, Heidelberg (1992)

Learning, Information Exchange, and Joint-Deliberation through Argumentation in Multi-agent Systems

Santi Ontañón¹ and Enric Plaza²

¹ CCL, Cognitive Computing Lab Georgia Institute of Technology,
Atlanta, GA 30332/0280
santi@cc.gatech.edu

² IIIA, Artificial Intelligence Research Institute - CSIC, Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia (Spain)
enric@iia.csic.es

Abstract. Case-Based Reasoning (CBR) can give agents the capability of learning from their own experience and solve new problems, however, in a multi-agent system, the ability of agents to collaborate is also crucial. In this paper we present an argumentation framework (AMAL) designed to provide learning agents with collaborative problem solving (joint deliberation) and information sharing capabilities (learning from communication). We will introduce the idea of CBR multi-agent systems ($\mathcal{M}AC$ systems), outline our argumentation framework and provide several examples of new tasks that agents in a $\mathcal{M}AC$ system can undertake thanks to the argumentation processes.

1 Introduction

Case-Based Reasoning (CBR) [1] can give agents the capability of learning from their own experience and solve new problems [2]. Moreover, in a multi-agent system, the ability of agents to collaborate is crucial in order to benefit from the information known by other agents. In this paper we will present an argumentation framework designed for learning agents (AMAL), and show that agents can use it to both (1) joint deliberation, and (2) learning from communication.

Argumentation-based joint deliberation involves discussion over the outcome of a particular situation or the appropriate course of action for a particular situation. Learning agents are capable of learning from experience, in the sense that past examples (situations and their outcomes) are used to predict the outcome for the situation at hand. However, since individual agents experience may be limited, individual knowledge and prediction accuracy is also limited. Thus, learning agents that are capable of arguing their individual predictions with other agents may reach better prediction accuracy after such an argumentation process.

Most existing argumentation frameworks for multi-agent systems are based on deductive logic or some other deductive logic formalism specifically designed to support argumentation, such as default logic [3]. Usually, an argument is seen as a logical statement, while a counterargument is an argument offered in opposition to another argument [4,14]; agents use a preference relation to resolve conflicting arguments. However, logic-based argumentation frameworks assume agents with preloaded knowledge

and preference relation. In this paper, we focus on an *Argumentation-based Multi-Agent Learning (AMAL)* framework where both knowledge and preference relation are learned from experience. Thus, we consider a scenario with agents that (1) work in the same domain using a shared ontology, (2) are capable of learning from examples, and (3) communicate using an argumentative framework.

This paper presents a case-based approach to address both: how learning agents can generate arguments from examples, and how can they define a preference relation among arguments based on examples. The agents use case-based reasoning (CBR) [1] to learn from past cases (where a case is a situation and its outcome) in order to predict the outcome of a new situation. We propose an argumentation protocol inside the AMAL framework that supports agents in reaching a joint prediction over a specific situation or problem — moreover, the reasoning needed to support the argumentation process will also be based on cases. Finally, we present several applications where the argumentation framework can be useful. First we will show how using argumentation agents can achieve joint deliberation, and we'll see how agents can act as committees or as an information market. Then we will show how agents can use argumentation as an information sharing method, and achieve effective learning from communication, and information sharing among peers.

The paper is structured as follows. Section 2 introduces our multi-agent CBR (MAC) framework. After that, Section 3 briefly describes our argumentation framework. Section 4 presents several applications of the argumentation framework, and finally Section 5 presents related work. The paper closes with related work and conclusions sections.

2 Multi-agent Case-Based Reasoning Systems

A *Multi-Agent Case Based Reasoning System (MAC)* $\mathcal{M} = \{(A_1, C_1), \dots, (A_n, C_n)\}$ is a multi-agent system composed of $\mathcal{A} = \{A_i, \dots, A_n\}$, a set of CBR agents, where each agent $A_i \in \mathcal{A}$ possesses an individual case base C_i . Each individual agent A_i in a MAC is completely autonomous and each agent A_i has access only to its individual and private case base $C_i = \{c_1, \dots, c_m\}$ consisting of a collection of cases. CBR methods solve new problems by retrieving similar problems stored in a case base, where each *case* is a previously solved problem. Once a set of problems has been retrieved, the solution to the problem at hand is computed by reusing the solution contained in the retrieved cases (adapting or combining those solutions if needed). The newly solved problem might be incorporated into the case base as another case.

Agents in a MAC system are able to individually solve problems by using case-based reasoning. In this paper we will limit our selves to analytical tasks, where solving a problem means to identify a particular *solution class* among a set of possible solutions. For example, diagnosing a patient with the right disease, classifying a customer in the right risk category for a loan, etc. CBR gives agents the capability to individually learn how to solve these kinds of tasks from experience, however, in a multi-agent system where each agent is exposed to different experiences we would like agents to collaborate and make use of information known by other agents. However, we are not interested in complete information sharing, but in a selective information sharing that only shares the information that is needed for the task at hand, thus keeping the amount of information

each agent knows and has to share manageable. The **AMAL** framework presented in this paper complements **MAC** systems by allowing agents to perform joint deliberation (solve classification tasks in a collaborative way) and learning from communication.

3 Argumentation-Based Multi-agent Learning: **AMAL**

The **AMAL** argumentation framework is based on the idea that CBR agents can justify the solutions they produce for new problems, and use those justifications as arguments. The kinds of arguments that CBR agents can generate are thus based on justifications and cases. In the following sections we will define the idea of justifications, then define the set of argument types that agents can use in the **AMAL** framework, after that we will introduce a preference relation based in cases, and finally present the **AMAL** argumentation protocol.

3.1 Justified Predictions

The basis of the **AMAL** framework is the ability of some machine learning methods to provide *explanations* (or *justifications*) to their predictions. We are interested in justifications since they can be used as arguments. Most of the existing work on explanation generation focuses on generating explanations to be provided to the user. However, in our approach we use explanations (or justifications) as a tool for improving communication and coordination among agents.

In particular in the **AMAL** framework agents use CBR as their learning and problem solving method. Since CBR methods solve problems by retrieving cases from a case base, when a problem P is solved by retrieving a set of cases C_1, \dots, C_n , the justification built will contain the relevant information from the problem P that made the CBR system retrieve that particular set of cases, i.e. it will contain the relevant information that P and C_1, \dots, C_n have in common. So, when an agent solves a problem providing a justification for its solution, it generates a *justified prediction*. A *Justified Prediction* is a tuple $J = \langle A, P, S, D \rangle$ where agent A considers S the correct solution for problem P , and that prediction is justified a symbolic description D .

3.2 Arguments and Counterarguments

For our purposes an *argument* α generated by an agent A is composed of a statement S and some evidence D supporting S as correct. In the context of **MAC** systems, agents argue about predictions for new problems and can provide two kinds of information: a) specific cases $\langle P, S \rangle$, and b) justified predictions: $\langle A, P, S, D \rangle$. Using this information, we can define three types of arguments: justified predictions, counterarguments, and counterexamples. A *justified prediction* α is generated by an agent A_i to argue that A_i believes that the correct solution for a given problem P is $\alpha.S$, and the evidence provided is the justification $\alpha.D$. A *counterargument* β is an argument offered in opposition to another argument α . In our framework, a counterargument consists of a justified prediction $\langle A_j, P, S', D' \rangle$ generated by an agent A_j with the intention to rebut an argument α generated by another agent A_i , that endorses a solution class S' different from that of $\alpha.S$ for the problem at hand and justifies this with a justification D' .

A *counterexample* c is a case that contradicts an argument α . Thus a counterexample is also a counterargument, one that states that a specific argument α is not always true, and the evidence provided is the case c that is a counterexample of α .

3.3 Case-Based Preference Relation

A specific argument provided by an agent might not be consistent with the information known to other agents (or even to some of the information known by the agent that has generated the justification due to noise in training data). For that reason, we are going to define a preference relation over contradicting justified predictions based on cases. Basically, we will define a *confidence* measure for each justified prediction (that takes into account the cases owned by each agent), and the justified prediction with the highest confidence will be the preferred one.

The idea behind case-based confidence is to count how many of the cases in an individual case base *endorse* a justified prediction, and how many of them are counterexamples of it. The more the endorsing cases, the higher the confidence; and the more the counterexamples, the lower the confidence. Specifically, an agent estimates the confidence of an argument as:

$$C_{A_i}(\alpha) = \frac{Y_{\alpha}^{A_i}}{1 + Y_{\alpha}^{A_i} + N_{\alpha}^{A_i}}$$

where $Y_{\alpha}^{A_i}$ are the set of cases in the case base of A_i that endorse α and $N_{\alpha}^{A_i}$ is the set of its counterexamples in the case base of A_i (see [10] for a more thorough explanation). Moreover, we can also define the *joint confidence* of an argument α as the confidence computed using the cases present in the case bases of all the agents in the group:

$$C(\alpha) = \frac{\sum_i Y_{\alpha}^{A_i}}{1 + \sum_i (Y_{\alpha}^{A_i} + N_{\alpha}^{A_i})}$$

In AMAL, agents use this joint confidence as the preference relation: a justified prediction α is preferred over another one β if $C(\alpha) \geq C(\beta)$.

3.4 The AMAL Argumentation Protocol

Let us present an intuitive description of the AMAL protocol (for a more formal description, see [10]). The interaction protocol of AMAL allows a group of agents A_1, \dots, A_n to deliberate about the correct solution of a problem P by means of an argumentation process. If the argumentation process arrives to a consensual solution, the joint deliberation ends; otherwise a weighted vote is used to determine the joint solution. Moreover, AMAL also allows the agents to learn from the counterexamples received from other agents.

The AMAL protocol consists on a series of rounds. At each round, each agent hold one single justified prediction as its preferred prediction. In the initial round, each agent generates its individual justified prediction for P and uses it as its initial preferred prediction. Then, at each round an agent may try to rebut the prediction made by any of the other agents. The protocol uses a token passing mechanism so that agents (one at a

time) can send counterarguments or counterexamples if they disagree with the prediction made by any other agent. Specifically, each agent is allowed to send one counterargument or counterexample each time he gets the token (notice that this restriction is just to simplify the protocol, and that it does not restrict the number of counterargument an agent can send, since they can be delayed for subsequent rounds). When an agent receives a counterargument or counterexample, it informs the other agents if it accepts the counterargument (and changes its prediction) or not (agents take that decision based on the preference relation and on incorporating counterexamples to their case base). Moreover, agents have also the opportunity to answer to counterarguments when they receive the token, by trying to generate a counterargument to the counterargument.

When all the agents have had the token once, the token returns to the first agent, and so on. If at any time in the protocol, all the agents agree or during the last n rounds no agent has generated any counterargument, the protocol ends. Moreover, if at the end of the argumentation the agents have not reached an agreement (an agreement is reached when the arguments that all the agents are holding at a particular round endorse the same solution), then a voting mechanism that uses the confidence of each prediction as weights is used to decide the final solution. Thus, **AMAL** follows the same mechanism as human committees: first each individual member of a committee exposes his arguments and discusses those of the other members (joint deliberation), and if no consensus is reached, then a voting mechanism is required.

Moreover, notice that agents can learn from the counterexamples received from other agents during an argumentation process. As we will show in the next section, counterexamples received by a particular agents are those ones that are in contradiction with the agent's predictions, and thus those ones that are useful to be retained.

4 Applications of **AMAL**

The **AMAL** argumentation framework gives agents in a **MAC** systems two new capabilities: joint deliberation and learning from communication. In this section we will present an evaluation of those two capabilities, in addition to a third evaluation where agents use **AMAL** as an “information sharing” mechanism.

4.1 Joint Deliberation

To evaluate the joint deliberation capabilities of agents using **AMAL** we designed the following experiment. A traditional machine learning training set is distributed among 5 agents without replication (the training set is split in 5 disjoint parts and each agent only has access to one of them). Then, one of the agents is asked to solve a new problem (not in the training set) and is asked to solve it. Such agent will engage in an argumentation process with some other agents in the system about the correct solution for the problem. We compare how accurate the prediction is using argumentation with respect to traditional voting mechanisms, and also study how much the number of agents that take part in the argumentation affects the prediction accuracy. We have made experiments in two different data sets: *soybean* (a propositional data set from the UCI machine learning repository) and *sponge* (a complex relational data set). The soybean data set has 307

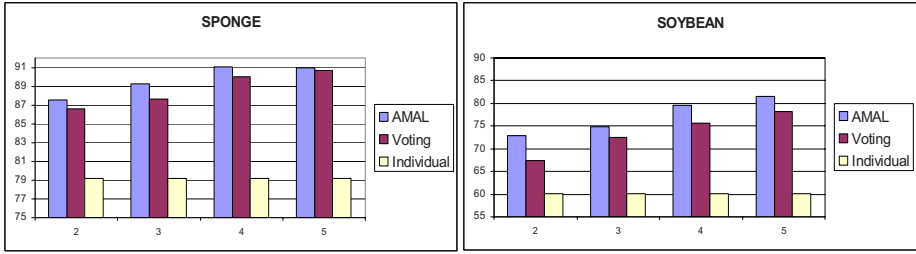


Fig. 1. Individual and joint accuracy for 2 to 5 agents

examples and 19 solution classes, while the sponge data set has 280 examples and 3 solution classes.

We ran experiments, using 2, 3, 4, and 5 agents respectively (in all experiments each agent has a 20% of the training data, since the training is always distributed among 5 agents). Figure 1 shows the result of those experiments. For each number of agents, three bars are shown: *individual*, *Voting*, and *AMAL*. The individual bar shows the average accuracy of individual agents predictions; the voting bar shows the average accuracy of the joint prediction achieved by voting but without any argumentation; and finally the *AMAL* bar shows the average accuracy of the joint prediction using argumentation. The results shown are the average of 5 10-fold cross validation runs.

Figure 1 shows that collaboration (voting and *AMAL*) outperforms individual problem solving. Moreover, as we expected, the accuracy improves as more agents collaborate, since more information is taken into account. We can also see that *AMAL* always outperforms standard voting, proving that joint decisions are based on better information as provided by the argumentation process.

For instance, the joint accuracy for 2 agents in the sponge data set is of 87.57% for *AMAL* and 86.57% for voting (while individual accuracy is just 80.07%). Moreover, the improvement achieved by *AMAL* over Voting is even larger in the soybean data set. The reason is that the soybean data set is more “difficult” (in the sense that agents need more data to produce good predictions). These experimental results show that *AMAL* effectively exploits the opportunity for improvement: the accuracy is higher only because more agents have changed their opinion during argumentation (otherwise they would achieve the same result as Voting).

4.2 Learning from Communication

Concerning learning from communication, we ran the following experiment: initially, we distributed a 25% of the training set among the five agents; after that, the rest of the cases in the training set is sent to the agents one by one; when an agent receives a new training case, it has several options: the agent can discard it, the agent can retain it, or the agent can use it for engaging an argumentation process. We compared the evolution of the individual classification accuracy of agents that perform each one of these 3 options. Figure 2 contains three plots, where NL (not learning) shows accuracy of an agent with no learning at all; L (learning), shows the evolution of the individual

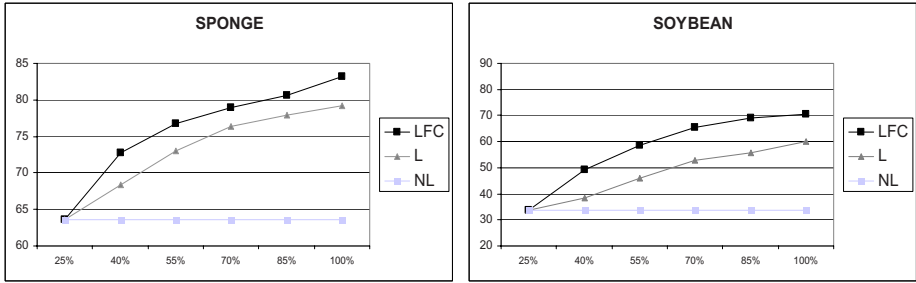


Fig. 2. Learning from communication resulting from argumentation in a system composed of 5 agents

classification accuracy when agents learn by retaining the training cases they individually receive (notice that when all the training cases have been retained at 100%, the accuracy should be equal to that of Figure 1 for individual agents); and finally LFC (learning from communication) shows the evolution of the individual classification accuracy of learning agents that also learn by retaining those counterexamples received during argumentation (i.e. they learn both from training examples and counterexamples received during argumentation).

Figure 2 shows that if an agent A_i learns also from communication, A_i can significantly improve its individual performance with just a small number of additional cases (those selected as relevant counterexamples for A_i during argumentation). For instance, in the soybean data set, individual agents have achieved an accuracy of 70.62% when they also learn from communication versus an accuracy of 59.93% when they only learn from their individual experience. The number of cases learnt from communication depends on the properties of the data set: in the sponges data set, agents have retained only very few additional cases, and significantly improved individual accuracy; namely they retain 59.96 cases in average (compared to the 50.4 cases retained if they do not learn from communication). In the soybean data set more counterexamples are learnt to significantly improve individual accuracy, namely they retain 87.16 cases in average (compared to 55.27 cases retained if they do not learn from communication). Finally, the fact that both data sets show a significant improvement points out the adaptive nature of the argumentation-based approach to learning from communication: the useful cases are selected as counterexamples, and they have the intended effect.

4.3 Information Sharing

prediction markets, also known as *information markets*. Prediction markets' goal is to aggregate information based on a *price signal* emitted by the members of a group. The advantage of the price signal is that it encapsulates both the information and the preferences of a number of individuals. In this approach, the task of aggregating information is achieved by *creating a market*, and that market should offer the right *incentives* for the participating people or agents to disclose the information they possess.

Table 1. Prediction markets accuracy with information exchange with varying number of acquaintances in the sponge dataset

<i>social network</i>	<i>market accuracy</i>	<i>individual accuracy</i>	<i>average reward</i>
<i>0 acquaintances</i>	89.71%	74.21%	10.35
<i>1 acquaintances</i>	90.57%	83.99%	11.42
<i>2 acquaintances</i>	91.29%	86.63%	12.14
<i>3 acquaintances</i>	91.14%	87.64%	11.94
<i>4 acquaintances</i>	91.07%	88.16%	11.85
<i>5 acquaintances</i>	91.21%	88.21%	11.93

Prediction Markets provide agents with an incentive to provide accurate predictions (since they receive some bonus if they provide the right answer), therefore, it is rational for agents to consult with other agents, before casting their votes. Thus, we can distinguish two phases: an information gathering phase, where agents consult with some of their acquaintances, and a joint deliberation phase, where agents cast their votes for particular solutions, together with a price signal (the price signal can be seen as how much money the agent bets into the predicted solution, and is proportional to the reward the agent will get if its prediction is correct).

In this experiment we will use **AMAL** as a framework for information sharing, and to evaluate it, we designed the following experiment. We will split the training set among 8 agents, and each agent in the system will have a small set of acquaintances with which it will share information before participating in the market. To perform information sharing, an agent will do the following: it will first generate its own individual prediction for the problem at hand using its local case base, and then it will start a one-to-one argumentation process with one of its acquaintances. The outcome of this argumentation is a more informed prediction than the original one. Using that prediction as a starting point, the agent will engage in another one-to-one argumentation process with its next acquaintance, and so on. After each argumentation process, the resulting prediction is stronger and stronger since it takes into account information known by more agents (without the agents having to share their case bases). The resulting prediction is casted by the agent as its vote in the prediction market, and the joint confidence (computed during the argumentation processes) of that prediction is used to compute his price signal (the higher the confidence, the higher the price signal).

We have performed experiments with 0 to 5 acquaintances and logged the prediction accuracy of the market, the prediction accuracy of each individual agent, and also the average money reward received by each agent per problem when agents can bet between 0 and 100 monetary units per problem, and all the agents that predicted the right solution split all the money that every agent bet (plus a 10% bonus).

Table 1 shows that information exchange is positive both for the individual agents and for the market as a whole. We can see that the more acquaintances an agent has, the higher its individual prediction. For instance, agents with 0 acquaintances have an accuracy of 74.21% while agents with 1 acquaintance have an accuracy of 83.99%, and when they have 5 acquaintances, their accuracy is increased to 88.21%. Moreover, the predictive accuracy of the market increases from 89.71% when agents do not perform information exchange, to above 91% when agents have more 1 acquaintances.

Concerning information exchange, the experiments show that individual and market accuracy improve. This means that the agents make a more informed prediction, and thus that AMAL is effective in providing agents with enough information to correct previously inaccurate predictions.

5 Related Work

Concerning CBR in a multi-agent setting, the first research was on “negotiated case retrieval” [12] among groups of agents. Our work on multi-agent case-based learning started in 1999 [7]; later Mc Ginty and Smyth [8] presented a multi-agent collaborative CBR approach (CCBR) for planning. Finally, another interesting approach is *multi-case-base reasoning* (MCBR) [6], that deals with distributed systems where there are several case bases available for the same task and addresses the problems of cross-case base adaptation. The main difference is that our MAC approach is a way to distribute the *Reuse* process of CBR (using a voting system) while *Retrieve* is performed individually by each agent; the other multi-agent CBR approaches, however, focus on distributing the *Retrieve* process.

Research on MAS argumentation focus on several issues like a) logics, protocols and languages that support argumentation, b) argument selection and c) argument interpretation. Approaches for logic and languages that support argumentation include defeasible logic [4] and BDI models [14]. Although argument selection is a key aspect of automated argumentation (see [13] and [14]), most research has been focused on preference relations among arguments. In our framework we have addressed both argument selection and preference relations using a case-based approach.

Finally, concerning argumentation-based machine learning, Fukumoto and Sawamura [5] propose a new theoretical framework for argumentation-based learning, where they focus on what is the belief status of an agent after receiving a new argument. The main difference with our work is that they perform a theoretical analysis of the belief revision problem after receiving an argument, where as we are concerned with the full problem of how to generate arguments, evaluate them, and learn from them, all based on learning from examples. Amgoud and Serrurier [2] propose an argumentation framework for classification where both examples and hypothesis are considered as arguments in the same way as in our framework. However, in their framework they focus on how to extract valid and justified conclusions from a given set of examples and hypothesis, where as in our framework we are concerned with how those hypothesis are also generated. Moreover, they only focus on the single agent situation. Other work has tried to improve the performance of machine learning methods by combining them with argumentation techniques. Možina et al. [9] where they introduce the idea of argued examples to improve the reduce the space of the hypothesis space and help producing more meaningful hypothesis.

6 Conclusions

In this paper we have presented an argumentation-based framework for multi-agent learning, AMAL, that allows a group of learning agents to perform joint deliberation

and information sharing. The main contributions of this work are: a) an argumentation framework for learning agents; b) a case-based preference relation over arguments, based on computing an overall confidence estimation of arguments; and c) an argumentation-based approach for learning from communication.

As future work, we plan to explore the situations where we have heterogeneous agents that use different learning methods to generate arguments, and we also plan to explore more realistic the effect of having non-trustable agents, that do not always reveal their truth information.

Acknowledgements. Support for this work came from projects MID-CBR TIN2006-15140-C03-01, and Agreement Technologies CSD2007-0022.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications* 7(1), 39–59 (1994)
2. Amgoud, L., Serrurier, M.: Arguing and explaining classifications. In: Rahwan, I., Parsons, S., Reed, C. (eds.) *Argumentation in Multi-Agent Systems*. LNCS (LNAI), vol. 4946, pp. 164–177. Springer, Heidelberg (2008)
3. Brewka, G.: Dynamic argument systems: A formal model of argumentation processes based on situation calculus. *Journal of Logic and Computation* 11(2), 257–282 (2001)
4. Chesñevar, C.I., Simari, G.R.: Formalizing Defeasible Argumentation using Labelled Deductive Systems. *Journal of Computer Science & Technology* 1(4), 18–33 (2000)
5. Fukumoto, T., Sawamura, H.: Argumentation-based learning. In: Maudet, N., Parsons, S., Rahwan, I. (eds.) *ArgMAS 2006*. LNCS (LNAI), vol. 4766, pp. 17–35. Springer, Heidelberg (2007)
6. Leake, D., Sooriamurthi, R.: Automatically selecting strategies for multi-case-base reasoning. In: Craw, S., Preece, A.D. (eds.) *ECCBR 2002*. LNCS (LNAI), vol. 2416, pp. 204–219. Springer, Heidelberg (2002)
7. Martín, F.J., Plaza, E., Arcos, J.-L.: Knowledge and experience reuse through communications among competent (peer) agents. *International Journal of Software Engineering and Knowledge Engineering* 9(3), 319–341 (1999)
8. McGinty, L., Smyth, B.: Collaborative case-based reasoning: Applications in personalized route planning. In: Aha, D.W., Watson, I., Yang, Q. (eds.) *ICCBR 2001*. LNCS (LNAI), vol. 2080, pp. 362–376. Springer, Heidelberg (2001)
9. Možina, M., Žabkar, J., Bratko, I.: Argument based machine learning. *machine learning* 171, 922–937 (2007)
10. Ontañón, S., Plaza, E.: Learning and joint deliberation through argumentation in multi-agent systems. In: *Proc. AAMAS 2007*, pp. 971–978. ACM, New York (2007)
11. Plaza, E., Ontañón, S.: Learning collaboration strategies for committees of learning agents. *Journal of Autonomous Agents and Multi-Agent Systems* 13, 429–461 (2006)
12. Nagendra Prasad, M.V., Lesser, V.R., Lander, S.: Retrieval and reasoning in distributed case bases. Technical report, UMass Computer Science Department (1995)
13. Sycara, K., Kraus, S., Evenchik, A.: Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence Journal* 104, 1–69 (1998)
14. Jennings, N.R., Parsons, S., Sierra, C.: Agents that reason and negotiate by arguing. *Journal of Logic and Computation* 8, 261–292 (1998)

A User-Centric Service Composition Approach

Gabriela Vulcu, Sami Bhiri, Manfred Hauswirth, and Zhangbing Zhou

Digital Enterprise Research Institute, National University of Ireland, Galway,

{firstname.lastname}@deri.org

<http://www.deri.ie>

Abstract. Frequently, in SOA, there exists no single service able to perform a given task, but there is a composition of services that can. In a continuously changing world, manual service composition is unrealistic and so is a fully automatic composition process: it is unreal to have a user specifying his request, pressing the magic button and waiting for the perfect solution. We believe that a certain degree of user involvement is required during the composition process while offering automatization when needed. This paper presents the overview of a user-centric approach for composing services into complex processes that are closest to the user's specific needs. We distinguish between two phases: the automatic composition at the service capability level and the user's contribution in refining the composition according to his specific requirements like domain specific QoS parameters at different granularity levels and the process structure.

Keywords: service composition, service capability, user-centric programming, workflow patterns.

1 Introduction

Frequently, in SOA, there exist no single service capable of performing a given task, but there is a combination of services that can. Given the big amount of services that exist and the more and more diversified needs nowadays, we cannot talk about manual composition of services in a continuously changing world. Thus automatization to support service composition is needed.

However, on the other hand a fully automatic composition process is also out of discussion: it is unreal to have a user specifying his request, press the magic button and then wait for the perfect solution. A certain degree of user involvement is required due to the fact that different users have different specific requirements at several layers of granularity.

In this paper we present a user-centric approach for composition of services. We believe that our approach is more realistic and practical than existing ones since it involves user during the composition process (to refine better the result according to his requirements) while offering the required automatization when needed.

The rest of this paper is structured as follows. Section 2 gives an overview of our approach. Section 3 introduces the conceptual model used as the basis for

the proposed solution. Section 4 presents some of the algorithms we developed. Section 5 states our approach against the state of the art and related work. Section 6 concludes.

2 Overview

In the following, we give an overview of our approach that proposes to automatically build offline an intermediate structure that will be used on run time (following a user request) for generating a composite service. The main guiding principles of our approach are:

1. *Capability-based composition*: different from existing solutions, our approach reasons on a capability level and not on a service level which reduces the search space since a capability can be achieved by more than one service.
2. *Offline automatic computing*: In order to offer a user acceptable time response we propose to use automatic reasoning offline for building an intermediate knowledge structure that will be used as a basis for dynamic composition. Using automatic reasoning is essential to deal with the relatively great number of services (from a human perspective) and their not so friendly user descriptions.
3. *User involvement*: It is hard that a user defines from the beginning in one shot all his needs. In addition, for the same requested business functionality, different users may require different properties at different level of granularity like QoS parameters and process structure. That is why we claim that it is more realistic and practical to allow user intervention during the composition process.

As depicted in Figure 1 we distinguish 5 computing steps: *Capability Composition Models* (a.k.a CCMs) *computing*, *CCMs marking*, *Capability Flow computing*, *User-Driven Composition* and *Service Selection*. The first two steps are achieved offline while the other three are achieved on run time following a user request and while user intervention.

The first step computes offline an internal knowledge structure called CCMs. Each capability may have a CCM that outlines 1 or many alternatives of how it (the capability) can be achieved by combining other capabilities. An alternative in a CCM is basically a workflow which we call Capability Flow (a.k.a. CF) which follows a block pattern structure.

CCMs have to respect a hierarchical decomposition. Informally put, a CCM *ccm* of a capability *c* contains the coarsest grain CFs able to achieve *c*; possibly englobing other CF at finer granularity levels by unfolding its component capabilities (unfolding a capability means replacing it by one of the CCM's alternatives). This internal knowledge structure is maintained in step 2 such that it keeps up to date the realizability of stored capabilities according to the availability of services.

The CCMs are used by the third step in order to return to the user a directly realizable CF (see definition in section 4) able to achieve the requested

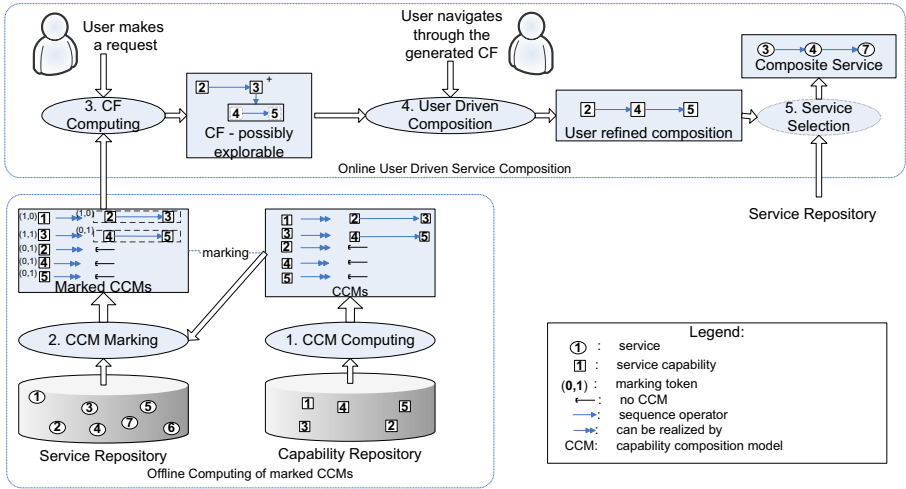


Fig. 1. An overview of our approach

capability. The returned CF can be further specified and refined by the user in step 4 by exploring eventually realizable capabilities (see definition in section 4) recursively. The result of step 4 is a user refined CF which plays the role of an abstract workflow that will be implemented by discovering and selecting the appropriate services to achieve the sub capabilities in step 5.

3 Conceptual Model

In this section we present the conceptual model that defines the main concepts we are considering as well as the relations between them. We also give the semantics of the block patterns used to define capability flows.

3.1 Underlying Model

Figure 2 shows the UML class diagram of our model. One central element of our model is *CapabilityCompositionModel* which consists of one or many *CapabilityFlows*. The *CapabilityFlowElement* (a.k.a. CFE) is an “umbrella” concept for the *Capability* and *CapabilityFlow* concepts and it is used in the recursive definition of the *CapabilityFlow*. The *Capability* concept denotes a business functionality and it may have 0 or 1 *CapabilityCompositionModel*. The *CapabilityFlow* concept is defined recursively as a block pattern workflow structure through the *Capability* concept and four block pattern concepts.

We define the following block pattern flows based on the workflow patterns defined by van der Aalst in [11]: (1) *Sequence* (a.k.a. SEQ) - a list of *OrderedElements*; an *OrderedElement* (a.k.a. OE) is a *CapabilityFlowElement* which has an order in the sequence it belongs to. (2) *ParallelSplitSynchronize* (a.k.a.

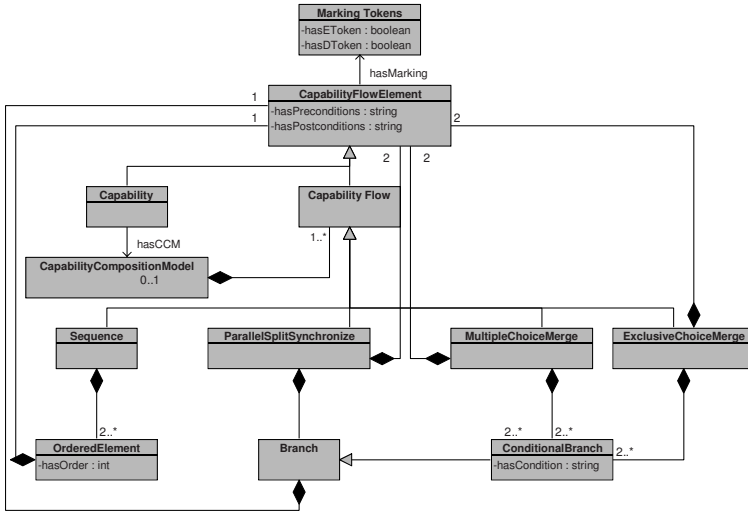


Fig. 2. The underlying model

PSS) - a pair of parallel split and synchronizing gateways . Practically it consists of *StartCFE* and *EndCFE* CFEs and a set of *Branches*, where a *Branch* (a.k.a. Br) consists of a CFE. (3) *Multiple Choice Merge* (a.k.a. MCM) - a pair of multi-choice and synchronizing merge gateways . Practically it consists of *StartCFE* and *EndCFE* CFEs and a set of *ConditionalBranches*, where a *ConditionalBranch* (a.k.a. Br(c)) is a *Branch* that fulfills a condition c . At runtime there can be many Br(c) active. (4) *Exclusive Choice Merge* (a.k.a. ECM) - a pair of exclusive choice and exclusive merge gateways . Practically it consists of *StartCFE* and *EndCFE* CFEs and a set of *ConditionalBranches*. At runtime only one Br(c) can be active.

A *CapabilityFlowElement* has preconditions and postconditions. Without loss of generality and for brevity reasons we assume that a *CapabilityFlowElement* preconditions and postconditions include its inputs and outputs respectively.

As mentioned in Section 2, step 2 uses marking of capabilities and CCMs. Thus we incorporate this in the model by the attribute *hasMarking* of type *MarkingTokens* of the *CapabilityFlowElement*. The *MarkingTokens* concept encapsulates two attributes : *hasDToken* and *hasEToken* to mark the fact that a CFE is directly or eventually realizable. These notions are explained in section 4.

We complement the above Class diagram by a set of OCL constraints in order to define more accurately and precisely the relation between a capability and CFs in its CCM. Due to lack of space we detail only when a CF, in general, can be considered as an alternative to a capability. Let c a capability and cf a CF, c can be replaced by cf iff the precondition of c entails the precondition of cf and the postcondition of cf entails the postcondition of c .

3.2 Semantics of Block Patterns

The semantic of a block pattern is defined by two parts: (a) the relation that must exist between its component elements and (b) the precondition and postcondition of the block pattern seen as a composite capability. First let's introduce some useful notation:

A and B denote any *CapabilityFlowElement*; Br , OE , $Br(c)$, with the condition c written in conjunctive normal form as defined in the previous subsection; $Prec(A)$ and $Post(A)$ refer to the precondition and postcondition of A respectively. The pre/postconditions of a Br and OE correspond to the pre/postconditions of its CFE. The pre/postcondition of a $Br(c)$ correspond to the conjunction between the pre/postconditions of its CFE and the branch condition c . $Prec_A(B)$ denotes the preconditions of B which are not satisfied by the postconditions of A . $Post_A(B)$ refers to the postconditions of B which are not contradicted by the postconditions of A . Table 1 summarizes the semantics of the used block patterns.

As an example we give the interpretation for the sequence pattern, $A = Seq(OE_1, OE_2)$; the others being defined in a similar way:

Relation between A's elements: the postconditions of OE_1 are a necessary but not obligatory a sufficient condition for the preconditions of OE_2 .

Table 1. The semantics of block patterns

Pattern	Relations between A's elements	Block pattern's description
Sequence $A = Seq(OE_1, OE_2)$	$Post(OE_1) \Leftarrow Prec(OE_2)$	$Prec(A) = Prec(OE_1) \wedge Prec_{OE_1}(OE_2),$ $Post(A) = Post_{OE_2}(OE_1) \wedge Post(OE_2)$
Parallel Split Synchronize $A = PSS(StartCFE, Br_1, Br_2, \dots, Br_n, EndCFE)$	$Post(StartCFE) \Leftarrow Prec(Br_i),$ $Post(Br_i) \Leftarrow Prec(EndCFE),$ $i = 1, n$	$Prec(A) = Prec(StartCFE) \wedge Prec_{StartCFE}(Br_1) \wedge Prec_{StartCFE}(Br_2) \wedge \dots \wedge Prec_{StartCFE}(Br_n) \wedge Prec_{Br_1, \dots, Br_n}(EndCFE) \wedge Prec_{StartCFE}(EndCFE),$ $Post(A) = Post(EndCFE) \wedge Post_{EndCFE}(Br_1) \wedge Post_{EndCFE}(Br_2) \wedge \dots \wedge Post_{EndCFE}(Br_n) \wedge Post_{Br_1, \dots, Br_n}(StartCFE) \wedge Post_{EndCFE}(StartCFE)$
Multiple Choice Merge $A = MCM(StartCFE, Br_1(c_1), Br_2(c_2), \dots, Br_n(c_n), EndCFE)$	$Post(StartCFE) \Leftarrow ((c_1 \wedge Prec(Br_1)) \vee (c_2 \wedge Prec(Br_2)) \vee \dots \vee (c_n \wedge Prec(Br_n))),$ $((c_1 \wedge Post(Br_1)) \vee (c_2 \wedge Post(Br_2)) \vee \dots \vee (c_n \wedge Post(Br_n))) \Leftarrow Prec(EndCFE)$	$Prec(A) = Prec(StartCFE) \wedge Prec_{StartCFE}(Br_i) \wedge Prec_{Br_i}(EndCFE) \wedge Prec_{StartCFE}(EndCFE),$ $Post(A) = Post(EndCFE) \wedge Post_{EndCFE}(Br_i) \wedge Post_{Br_i}(StartCFE) \wedge Post_{EndCFE}(StartCFE),$ for each branch where c_i evaluates to true.
Exclusive Choice Merge $A = ECM(StartCFE, Br_1(c_1), Br_2(c_2), \dots, Br_n(c_n), EndCFE)$	$Post(StartCFE) \Leftarrow ((c_1 \wedge Prec(Br_1)) \otimes (c_2 \wedge Prec(Br_2)) \otimes \dots \otimes (c_n \wedge Prec(Br_n))),$ $((c_1 \wedge Post(Br_1)) \otimes (c_2 \wedge Post(Br_2)) \otimes \dots \otimes (c_n \wedge Post(Br_n))) \Leftarrow Prec(EndCFE),$ where c_1, c_2, \dots, c_n are disjunctive.	$Prec(A) = Prec(StartCFE) \wedge Prec_{StartCFE}(Br_i) \wedge Prec_{Br_i}(EndCFE) \wedge Prec_{StartCFE}(EndCFE)$ $Post(A) = Post(EndCFE) \wedge Post_{EndCFE}(Br_i) \wedge Post_{Br_i}(StartCFE) \wedge Post_{EndCFE}(StartCFE),$ for the only branch Br_i where c_i evaluates to true.

Block pattern's description: The preconditions of A are the preconditions of OE_1 and preconditions of OE_2 which are not satisfied by the postconditions of OE_1 . The postconditions of A are the postconditions of OE_2 and the postconditions of OE_1 which are not contradicted by postconditions of OE_2 .

4 Algorithms

In this section we focus on the algorithms for achieving step 2 and step 3 presented in Figure 1. Before going on presenting them, let's introduce some useful notions.

Depending on the availability of stored services, we can classify CFEs as follows: (1) A capability c is said to be **directly realizable** if it exists at least one service already hosted able to achieve it. (2) A capability flow CF is said to be **directly realizable** if all of its component capabilities are directly realizable. (3) A capability c is said to be **eventually realizable** if it exists a directly realizable or eventually realizable CF able to achieve it. (4) A capability flow CF is said to be **eventually realizable** if it is not directly realizable and each of its component capabilities is either directly or eventually realizable.

4.1 The *mark* Algorithm

The *mark* algorithm is used in step 2 and it enables qualifying if CFEs are realizable or not. We use two tokens to mark CFEs, as mentioned in the previous section: the *hasDToken* specifies if the CFE is directly realizable while the *hasEToken* specifies if the CFE is eventually realizable.

The *mark* algorithm (Algorithm 1) is a recursive algorithm and it applies to each CFE. The marking of a capability depends on the marking of the alternative CFs of its CCM and on the availability of services that are able to achieve it. Thus *hasDToken* of a capability c is set to 1 if there is at least one service that can achieve c , otherwise it is 0 (line 5.). The *hasEToken* of a capability c is set to 1 (line 11.) if the CCM of c contains at least one CF that has the *hasDToken* or the *hasEToken* set (line 10.).

The marking of a CF depends on the marking of its constituent capabilities and on the block pattern that defines it. Thus for a CF which follows a sequence pattern, the *hasDToken* is set to 1 only if the *hasDToken* of all constituent capabilities is 1 (line 16.). The *hasEToken* of a sequence pattern will be set to 1 if the *hasDToken* is 0 and all its constituent capabilities have either the *hasDToken* or *hasEToken*, or both set to 1 (line 18.). In case of a PSS, MCM or ECM, the *hasDToken* is set to 1 if the *hasDToken* of *StartCFE*, *EndCFE* and of each branch/ conditional branch is 1 (line 16.). The *hasEToken* of PSS, MCM, ECM is set to 1 if the *hasDToken* is 0 and the *StartCFE*, *EndCFE* and each branch/ conditional branch have the *hasDToken* or *hasEToken*, or both set to 1 (line 18.). The marking of *OE*, *Br* or *Br(c)* (line 15.) is basically the marking of their constituent CFE.

A capability c can be achieved by a set of services. In a dynamic environment like the one we consider, the direct realisability of a capability can change and

therefore affect the marking of other capabilities and CFs. Therefore we provide a way to maintain the marking of CFEs consistent with the current changes by propagating ‘upwards’ (i.e. from the changed capability to the CCMs it belongs to) the computing of tokens of all CFEs that might have been influenced by the change.

Algorithm 1. The *mark* Algorithm

Input: *CFE*: *cfe*: the CFE to be marked (either capability or CF)

Output: *MarkingTokens*: *markingTokens*: the result

```

1 begin
2   ebool, dbool ← false
3   markingTokens ← null
4   if cfe is a Capability then
5     markingTokens.hasDToken ← (cfe is directly realizable)?1:0
6     markingTokens.hasEToken ← 0
7     if cfe has CCM, ccm then
8       foreach cf of ccm do
9         tokens ← mark(cf)
10        if tokens.hasDToken || tokens.hasEToken then
11          markingTokens.hasEToken ← 1
12          break
13    else if cfe is a CapabilityFlow then
14      foreach OE or Br or BRc of cfe do
15        tokens ← mark(OE or Br or Br(c))
16        if !tokens.hasDToken then
17          dbool ← true
18          if !tokens.hasEToken then
19            ebool ← true
20      markingTokens.hasDToken ← !dbool
21      markingTokens.hasEToken ← !ebool
22    return markingTokens
23 end

```

4.2 The *explore* Algorithm

Conceptually we remark that each capability c has attached an hierarchical structure, lets call it capability tree denoted T_c which captures how c can be described and achieved at different levels of granularity. The root node of T_c is the capability c . The level $i + 1$ details more how c can be achieved by unfolding capabilities at the level i . Unfolding a capability means replacing it by one CF from its CCM .

Algorithm 2. The *explore* Algorithm.**Input:** CFE: *cfe*: the CFE to be explored (either capability or *CF*)**Output:** a directly realizable *CF* able to achieve the *cfe*

```

1 begin
2   if hasDToken of cfe is 1 then
3     return entity
4   else if hasEToken of cfe is 1 then
5     if cfe is Sequence(OE1,OE2) then
6       return Sequence(explore(OE1),explore(OE2))
7     else if cfe is a PSS/MCM/ECM(StartCFE,
8       Br1/Br1(c1),...,Brn/Brn(cn), EndCFE) then
9       return PSS/MCM/ECM(explore(StartCFE),
10        explore(Br1/Br1(c1)),...,explore(Brn/Brn(cn)),
11        explore(EndCFE))
12     else if cfe is a Capability then
13       intermediateCF ← choose a CF from cfe's CCM that is directly
14         realizable, or, if not, eventually realizable
15       return explore(intermediateCF)
16   return 'The request cannot be achieved.'
```

The *explore* algorithm (Algorithm 2) belongs to step 3 and it returns, for a given capability *c*, a directly realizable CF able to achieve *c*. The main idea is to recursively explore the capability's tree T_c , when necessary, to find out a directly realizable CF able to achieve the requested capability. The exploration will avoid the sterile paths (i.e. a sterile path is a path in the capability tree where its root is a capability which is neither directly nor eventually realizable - marking (0,0)) due to the marking of CFEs.

The *explore* algorithm will generate a directly realizable CF which will be presented to the user. If the capabilities in the generated CF are also eventually realizable, the generated CF can be further explored by the user in order to add QoS parameters to different granularity levels and to define the final structure of the composition. The exploration of *OE*, *Br* and *Br*(*c*) is the exploration of their constituent CFE.

5 Related Work

Representative work regarding service composition have been done considering two main directions: automatic service composition and manual service composition.

Among the automatic service composition approaches we mention: graph-based techniques [2, 3], classical AI-planning based techniques (forward-chaining - [4, 5, 12]; backward-chaining - [6]; estimated-regression - [7]) and non classical AI

planning techniques (HTN planning (hierarchical task networks) - [8], [12]). The advantage of these approaches is that the composition can be adaptable without too much effort since the work is done automatically by machine, thus providing a shorter time-to-market of new created services. The problem is that the generated plans represent simple structures like sequences or partial ordered plans that are not as expressive as workflows are and, usually, do not correspond with the real processes. AI plans, as they currently exist, are very simplified types of workflows containing only the sequence, AND-split and AND-join workflow patterns. Another inconvenience is that they do not allow customization of the composite service according to the user's specific needs. The user has to specify everything he want in one shot, when defining the goal of his request.

From the manual service composition context there are representative workflow-based approaches: model driven service composition [10], [13], [14] and process templates [9], [11]. Modeling services composition with languages like: BPML, BPMN, BPEL4WS, BPSS, WSCI is much closer to the real complex processes. Although a manual approach for service composition allows the user to customize the composite service as much as he wants, this approach is not scalable and does not adapt properly to the rapid changes nowadays when many services are created daily.

There are also hybrid approaches that consider a combination of the two described above. The European project SUPER [15] applies such solution for service composition by automatically composing individual tasks (goals) predefined earlier in a high-level composition model. Our approach offers more flexibility to the user who can intervene and specify his requirements in terms of QoS and composition structure. In addition, in SUPER the composition model might not be realizable with the existing stored services. In the contrary, our approach avoids such a situation by taking into account services availability.

6 Conclusion

In this paper we presented an approach for semantic web service composition in user-centric context. We believe that it is a realistic and practical approach which involves the user during the composition process to best fulfill his request. Among the advantages of our approach we mention: time saving due to the use of an internal knowledge structure which is computed in background and always maintained to reflect the availability of services. Moreover capability-based composition reduces search space due to the fact that a capability can achieve more than one hosted service.

References

1. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns (2003)
2. Zhang, R., Arpinar, I.B., Aleman-Meza, B.: Automatic Composition of Semantic Web Services. In: Proc. of the 2003 Int. Conf. on web Services (ICWS 2003) (2003)

3. Mao, Z.M., Brewer, E.A., Katz, R.H.: Fault-tolerant, Scalable, Wide Area internet Service Composition. In Technical Report UCB//CSD-01-1129, University of California, Berkley, USA (2001)
4. Ponnekanti, S., Fox, A.: SWORD: A Developer Toolkit for Web Service Composition. In: Proc. of the 11th Int. WWW Conf (WWW 2002), Honolulu, HI, USA (2002)
5. Martinez, E., Lesperance, Y.: Web Service Composition as a Planning Task: Experiments with Knowledge-Based Planning. In: Proc. of the 14th Int. Conf. on Automated planning and Scheduling (ICAPS 2004), Whistler, BC, Canada (2004)
6. Sheshagiri, M.: Automatic Composition and Invocation of Semantic Web Services. In Master's thesis University of Maryland, Baltimore County, USA (2004)
7. McDermott, D.: Estimated-Regression Planning for Interactions with Web Services. In: Proc. of the 6th. int. Conf. on Artificial Intelligence Planning Systems (AIPS 2002), Toulouse, France (2002)
8. Wu, D., Sirin, E., Hendler, J., Nau, D., Parsia, B.: Automatic DAML-s Web Service Composition using SHOP2. In: Fensel, D., Sycara, K.P., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, Springer, Heidelberg (2003)
9. Aggarwal, R., Verma, K., Miller, J., Milnor, W.: Dynamic Web Service Composition in METEOR-S. In Technical Report, LSDIS Lab, C.S. Dept. UGA (2004)
10. Orriens, B., Yang, J., Papazoglou, M.P.: Model Driven Service Composition. In: Proc. of the 1st Int. Conf. on Service Oriented Computing (2003)
11. Sirin, E., Parsia, B., Hendler, J.: Template-based Composition of Semantic Web Services. In: AAAI Fall Symposium on Agents and Semantic Web, Virginia, USA (2005)
12. Klusch, M., Gerber, A., Schmidt, M.: Semantic Web Service Composition Planning with OWLS-Xplan. In: 1st Int. AAAI Fall Symposium on Agents and Semantic Web (2005)
13. Grnmo, R., Skogan, D., Solheim, I., Oldevik, J.: Model-driven Web Services Development. In: Proc of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE 2004) (2004)
14. Skogan, D., Grnmo, R., Solheim, I.: Web Service Composition in UML. In: Proceedings of the Enterprise Distributed Object Computing Conference, Eighth IEEE International (2004)
15. SUPER: Semantics Utilized for Process Management within and Between Enterprises: <http://www.ip-super.org/>

Towards Reliable SOA – An Architecture for Quality Management of Web Services

Ingo J. Timm¹ and Thorsten Scholz²

¹ Goethe-University Frankfurt, Institute of Computer Science
Robert-Mayer-Str. 11-15, 60325 Frankfurt/Main, Germany

² University of Bremen, Center for Computing Technologies (TZI)
Am Fallturm 1, 28359 Bremen, Germany
timm@cs.uni-frankfurt.de, scholz@tzi.uni-bremen.de

Abstract. Service-oriented architectures are a new paradigm for business information systems. Providing flexible interfaces, dynamic reconfiguration of software systems, i.e., business information systems, becomes possible during runtime. However, there are critical issues in quality management of the resulting systems. In this paper we will discuss challenges in testing resp. ensuring quality of service of dynamically orchestrated systems and present approaches to automated certification of service-oriented architectures.

Keywords: quality of service, unit testing, certification management.

1 Introduction

The business trends of the last decade, like globalization, decreasing in-house production depth, and outsourcing, are leading to high demands on flexibility and competitiveness of enterprises. Accompanying, the need for flexible IT infrastructure on the software as well as on the hardware level increases significantly to enable enterprises to establish production networks real-time. Doing so, the integration of intra- and inter-organizational information systems is required. In the last years, web services and service-oriented architectures (SOA) have been propagated as an adequate means for these requirements, as they integrate description (UDDI), retrieval and publication (WSDL) as well as middleware (SOAP) on a standardization level.

Our vision of “Emergent Virtual Enterprises” combines flexible enterprise networks and software architectures. The evolving process starts with the customer’s demands which lead to a product specification. The following step creates a dynamic consortium of enterprises which represent the required production network, such that a virtual enterprise emerges. This consortium will implement the necessary process steps, e.g., product specification, manufacturing, assembling, disposition [1]. This evolving process is based on ideas on multiagent theory and should use principles of electronic market places (cf. Fig. 1). Of course, technical aspects like interoperability and performance are only one important issue for “Emergent Virtual Enterprises”. Aspects like trust, experience, emotions, and culture are a mandatory consideration, too, but will not be discussed in this paper. The technical configuration of production networks requires the integration of information flows including identification,

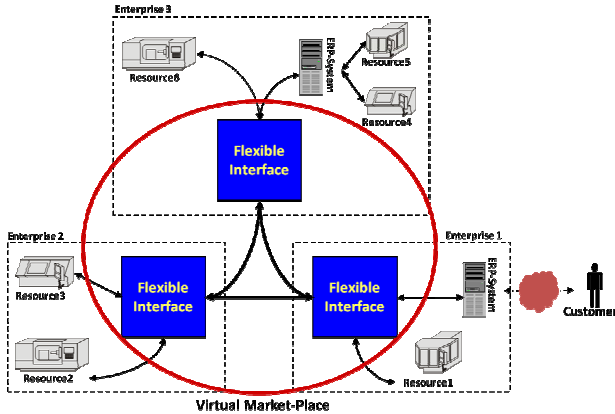


Fig. 1. Emergent Virtual Enterprises

selection, combination, and use of services. For real-life applications, the assessment of quality of services (QoS) is essential. In common approaches, QoS is defined by a multi-dimensional rating matrix containing objective as well as subjective parameters. Berbner et al. propagate that these parameters have to be identified by experts [2]. Another approach to manage QoS is based on contracts defining service-level-agreements with specific criteria. Both approaches include a high degree of human involvement and are consequently inadequate for emergent virtual enterprises.

On a more technical level, quality management of SOA, esp. in high dynamic environments, is not sufficiently solved. The challenge here is increased dramatically, if statefull web services or (intelligent) agents are considered. However, functionality and performance of the overall system depends on the individual quality of service as well as the interaction of services. In contrast to conventional systems, QoS may differ during runtime with respect to network load, service usage, or substitution of services resp. underlying algorithms. To guarantee reliability of a SOA there are two major views on the quality management for web services to consider: the perspective of the service provider and the perspective of the service consumer. The provider on the one hand has to ensure quality of his service even in unknown application contexts. Furthermore, there are specific interaction rules to be met. The consumer on the other hand has to select services in a concrete application context by individual quality statements. Obviously, a consumer has not the capacity or the knowledge to evaluate any available service. Therefore, a trusted entity is required, which provides evaluation of services for a group of service consumers. The quality management can be defined in three steps here: validation of implementation model (service provider), validation of design model (service provider), and validation of requirements model (service consumer resp. trusted entity).

The paper is organized as follows: in the second section, foundations of web service, SOA and quality management are introduced. Our architecture for quality management of web services is detailed out in section three. In the following, related work and consequences of our approach are discussed.

2 Rationale and Related Work

2.1 (Semantic) Web Services

Service retrieval, service chaining, and service application are primary research topics in the semantic web community; the underlying idea here is to include explicit semantics and automated reasoning capabilities [3], [4]. A key issue is the representation of functionality, skills, and capabilities [5]. Assessing quality of services is crucial to the validation process with respect to the requirements model, i.e., for implementing the acceptance test. Quality measures are multi-dimensional and are expressed in a QoS model (e.g. [6], [7]). These models distinguish between domain dependent dimensions, specifying specific business criteria, and generic quality criteria, specifying, e.g., execution price, response time, reliability, and reputation of the service and are based on an organizational viewpoint taken in, e.g., the work of [8]. Specific QoS criteria are identified in [9], where the author distinguishes four dimensions of QoS: run-time related, transaction support related, configuration management and cost related, and security related issues.

Retrieving a service to solve a specific problem implies the existence of a formal description of (a) the offered service capabilities and (b) the given problem. To provide the dynamic selection of distributed services to solve a problem, a standardized protocol to access services is needed. The W3C defines web service architectures and protocols, e.g., the description language WSDL to define a web service framework. Describing the capabilities of web services, with respect to using this information for dynamic inference processes, formal description languages, e.g., DAML-S and OWL-S are needed. However, these static descriptions of service capabilities limit the dynamic aspects distributed problem solving essentially. Work on capability management in the multiagent system community focuses on inferences on capabilities in order to enable higher applicability of a provided service (e.g. [10], [11]) and thus allowing for enhanced dynamic behavior of the overall system. Based on a representation of service capabilities, infrastructures like catalogues or yellow pages are used for match-making of service consumers and service providers.

2.2 Quality Management

Quality management of software systems should ensure that the system meets required properties by testing, validating, or verifying system's behavior. In distributed systems, quality management is more complex, esp. because of concurrency or inherent conflicts of interest in the case of different shareholders. In contrast to conventional software engineering with quality management preceding roll-out, in distributed systems, esp. in SOA, quality management is essential in runtime, too. Riedemann defined important quality criteria for distributed systems [12]: functionality (i.e., adequateness, correctness, interoperability, normative adequacy, and security) dependability (i.e., maturity, fault tolerance, and recovery), usability, efficiency, and adaptability. Following Riedemann, quality management combines constructive and analytic activities [12], while Menzies & Pecheur focus on the amount of expertise involved to classify different steps of quality management: testing, run-time monitoring, static analysis,

model checking, and theorem proving [13]. There are no specific testing methods available for SOA, but conventional testing methods can be applied, if i.e., special programs are used for simulating input sequences. Testing as a method is easy to use but exhaustive coverage of all possible cases is too costly. In runtime monitoring, a controlled environment is provided for evaluation of valid system's behavior. Here, a greater coverage is reached in comparison to testing but a formal model is needed. Static analysis is a code-based validation method for identification of error patterns in compiled programs. The method is capable of recognizing known error patterns with high accuracy but has its limitations if major error patterns are unknown. In model checking, an abstract model of the system as well as of the valid system states is required. It is based on the semantic level and therefore is in need of a formal requirements specification. Theorem proving is a logic-based method which tries to prove program correctness formally, is extremely complex and requires high degree of expert knowledge.

Various approaches to quality management of SOA focus on the organizational level using governance or service-level agreements only. The discussion on the need for specific validation methods for SOA is comparable to the early days of the object-oriented paradigm (OOP). Booch, for example, did not explicitly include validation methods in his methodology [14] while Rumbaugh et al argued that OOP by itself would reduce validation efforts [15]. The latter argumentation was disproved by studies [16]; reasons can be found in increased interdependencies of modules [17]. Problems occur as not every use case of objects is known in design-time such that implementations have to consider universal inputs which cannot be predicted or tested sufficiently [18]. Considering OOP, Sneed & Winter have postulated that coverage of tests is comparable low with respect to conventional software [19]. SOA can be seen as refinement of OOP with respect to information hiding and interoperability such that the outlined problems apply here, too.

3 Conceptual Framework

On an abstract level, software engineering of complex systems requires the development of at least three different models: requirements, design, and implementation model. Quality management has to ensure, that each of these models is valid [20]. In case of web services, quality management is two-fold: On the one hand, the service provider has to make sure, that the service show reliable behavior even in unknown environments and meet its interaction requirements. On the other hand, the service consumer is looking for that service which has the *best* QoS evaluation with respect to consumer's requirements. As long as any of the models are developed within a single organizational frame, the development of SOA does not have to be different from conventional software development. However, if service provider and service consumer are not contained in one organizational frame and the development of the provider has been independent from the consumer, the conventional approach is not adequate.

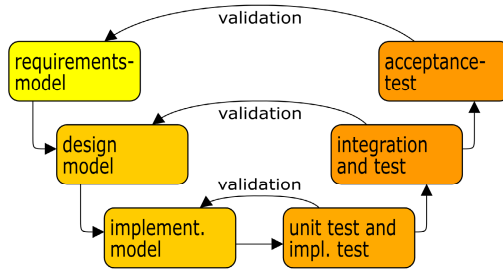


Fig. 2. Conceptual model of the quality management architecture

In our architecture (cf. Figure 2) we propose that the service provider validates the design and implementation model using adapted methodologies of unit tests and (simulation-based) runtime monitoring. For the service consumer we introduce a third-party-based certification approach for validating the requirements model. The proposed architecture contains such quality management methodologies which should be applied additionally to the conventional validation methods. In the following paragraphs we will outline three methodologies for quality management of web services.

3.1 Quality Management for the Implementation Model

The quality management for the implementation model is based on the assumption, that conventional quality management has been applied to the implementation model. The interaction between different objects in object-oriented programming is realized by method signatures and invocation. The signatures ensure valid invocations – depending from programming languages such interfaces are forced to be correct in the compilation step. In this aspect, SOA differs significantly: web services provide a unique interface for interoperability, i.e., any functionality is called by the same interface. The interface – on a structural level – is forced to be correct in the compilation. Calling different functionalities of a service requires messages with different content to be sent over the same interface. Therefore, a web service has to comply with different interaction patterns for the specified service.

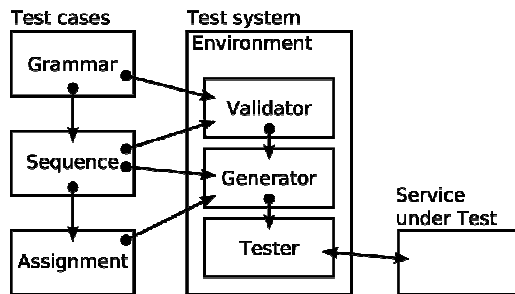


Fig. 3. Framework for tests of service interaction

On an implementation level, it is necessary to specify required behavior and testing procedures. To guarantee runtime behavior, unit tests are one of the dominating approaches in testing. Unit tests follow the idea to specify the requirements as test cases in the programming language, such that they can be executed automatically. The specified test cases can be used as a basis for documentation. On an abstract level, unit tests are a black-box testing method, as the specification of test cases is only in need of exported interface information. The motivation to transfer unit test to web service interaction is to specify interaction patterns for the web service in a executable way. Doing so, the specification allows for automated evaluation of the service and to ensure the usability and interoperability of this service for clients. The unit test operates as a client of the service and analyses the reaction of the service in the specified scenarios. This methodology and the resp. a test system (cf. Fig. 3) were introduced for testing of agent interaction on a syntactical level [21]. A grammar restricts the set of possible tests and limits the specifiable test cases to interaction protocols between client and server. A test scenario consists of a sequence of interaction messages with different parameter assignments, such that the execution of equal interaction sequences with different parameters is possible.

3.2 Quality Management for the Design Model

In analogy to testing, run-time monitoring of web services is based on the assumption that conventional validation of the design model has taken place. In conventional run-time monitoring, a controlled environment with a valid behavior is implemented. In the following, the system is supervised continuously in its application environment. In case of failure, detailed protocols are used for identifying and analyzing possible causes. The development of web services differs from conventional software engineering, as the application environment is usually not known and the SOA context is inherently distributed. Therefore, we propose to use a combined approach of simulating expected environments and run-time analysis. In our approach, a service or a service-based application is placed in a testing environment of different other services and consumers which are interacting with the service (cf. Fig. 4).

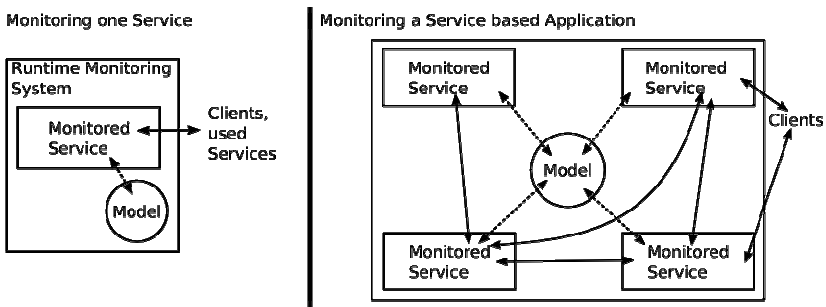


Fig. 4. Run-time monitoring in the SOA context

This validation scheme is similar to the problems of validating intelligent software agents. For the validation of intelligent agents or the validation of multiagent systems, we have applied simulation-based runtime monitoring, where a controlled environment as well as a control structure are established. Additionally, a specific agent is introduced in the multiagent system, which is collecting and documenting information about the dynamic system behavior. To cope with the challenge of great variations and unforeseeable states of environment, a stochastic simulator is introduced that triggers the control unit and establishes varying conditions in a high number of sequential runs. By this, the test procedure becomes a stochastic process and statistical analysis of the system behavior is possible, which is considered as a grey-box test. Statistical analysis is focused on both aspects, the dynamical behavior looking at parameter development over time (time series analysis) and summarizing results (reached state at the end of the run or mean values). Special statistical methods fit to these tasks, for example t-test, Kruskal-Wallis-test, ANOVA (analysis of variance) [20].

3.3 Quality Management for the Requirements Model

The first two models (implementation and design) have been discussed from the service provider perspective. In the presented solutions, details of the implementation could be considered within quality management. The perspective of a service consumer differs from conventional user perspective in software engineering significantly. The user is integrated in the requirements analysis step while in SOA the consumer is retrieving a service which is capable to solve a task. As requirement analysis has to be performed by the consumer, the consumer has to validate the requirements model, too. For practical reasons, it can be doubted, that any service consumer will have the capability or capacity to apply validation methods to any provided services relevant for him. The service provider will also perform quality management on the requirements model but with conventional validation methods.

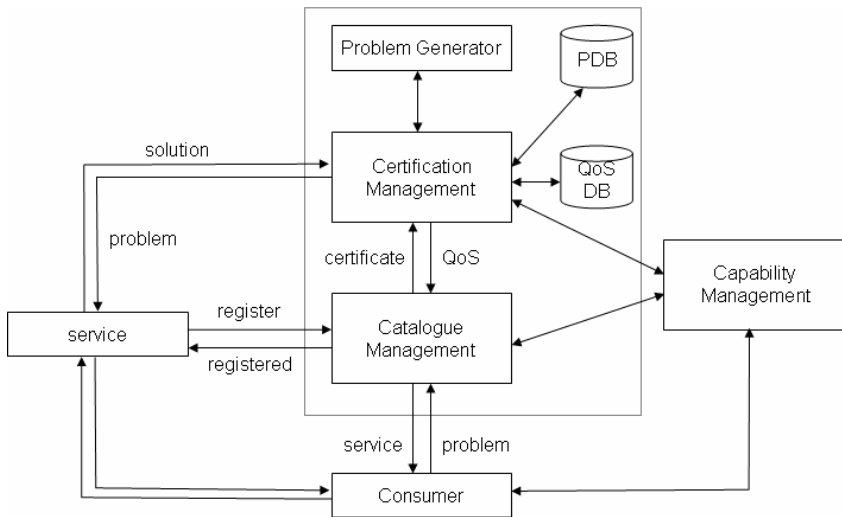


Fig. 5. Architecture for dynamic service certification

We propose certification management as a third-party approach for quality management of web services from a consumer perspective, which has been introduced in [5]. In this approach, a trusted entity is responsible for continuous assessment of web services. As the retrieval of web services depends on the current requirements of the consumer, it is necessary to explicitly allow for different QoS preferences. Three main components are mandatory:

- Capability management for the identification of the best-fitting service, i.e. services are identified on basis of their capabilities. The problem here is to match a given task to capabilities; in real-life applications this includes not only direct but also fuzzy mappings [11].
- Certification management for the evaluation of the services, i.e. available services need to be evaluated according to the offered capabilities as well as other quality issues resulting in a quality measurement.
- Catalogue management for supporting the service retrieval process, i.e. it integrates capability and certification management for providing a unified service exploration interface for service consumers.

The general idea of the certification architecture (cf. Figure 5) is comparable to unit testing: a trusted entity specifies requirements for specific capabilities of services. Therefore, standard problems or problem classes are defined, which can be used for the QoS assessment of concrete web services. The services are certified right after their registration with the catalogue and capability management. Consumers request services by specifying expected QoS and functionality at the catalogue management which filters and prioritizes available services. Doing so, the capability management derives by logical inferences services or service orchestrations which are meeting the requirements. Finally, the consumer provides feedback to the catalogue-service. If quality requirements were not met by the service, certification for this service is invoked.

This architecture has been implemented and evaluated with significant results within a simplified application domain (search) [5]. On the basis of these results we assume, that certification is beneficial if capabilities and performance can be specified formally.

4 Related Work

QoS is an important issue in research on quality management in SOA. Sheth et al. propose a service-oriented middleware which has an integrated QoS model supporting its automatic computation [7]. The QoS measure is proposed to be fine-tuned by simulation of changes in a workflow. This approach defines a model which lacks an explicit representation of service capabilities and focuses on standard issues like time, cost, and reliability and therefore leaves the question unanswered, whether a service is capable of solving a problem. In [22], the authors propose a QoS model, distinguishing between generic and domain dependent quality criteria. The model is used in a QoS registry, where web services are registered with a QoS based on reputation and user feedback. This approach also lacks a formal representation of service capabilities, and focuses on the basic criteria like execution time and pricing. Ran defines a

model for web service discovery with QoS, by introducing a certification component to the common UDDI architecture [9]. The author presents a broad, informal model for QoS, including a completeness measure. Nevertheless, the approach defines a static approach to certification and does not define its process. In the multiagent system community, Rovatsos et al. define a communication based performance measure for self diagnosis [23]. The authors define a generic model for measuring the performance of agent communication actions which can be used for self-repairing and self-optimizing, which is, to some extent a QoS measure. However, this approach focuses on measuring communication efforts and is not suited for measuring QoS. A different approach to reliable behavior of distributed services is trust (e.g. [24], [25]). In these approaches, trust models, based on, e.g., a service consumer community, is used for assessing the reliability of a service. These approaches share some ideas of our group concept in certification but they lack representation of service capabilities and certification mechanisms.

5 Discussion and Future Work

In this paper we introduced a conceptual framework for quality management of dynamically orchestrated software systems based on the combination of three quality management methodologies. The sequential application of each methodology focusing on one specific quality management problem of service-oriented software systems already shows promising results for an overall increase of the QoS – with respect to our experimental results [5,20]. However, with the conceptual framework we aim towards a seamless integration of these methodologies which is supposed to be more than just a sequence. In our future work, we will focus on the integration of these methodologies which will lead to a process-driven quality management encompassing design-time, implementation and run-time of distributed applications. Additionally, our future work will be focused on the application of the framework in real-life scenarios, e.g. dynamic supply-chain management in large manufacturing scenarios [26], where the complexity of the underlying software systems and the interaction processes requires mandatory and sophisticated quality management.

References

1. Tönshoff, H.K., Woelk, P.-O., Timm, I.J., Herzog, O.: Emergent Virtual Enterprises. In: Ueada, K. (ed.) *Proc. of the 4th International Workshop on Emergent Synthesis (IWES 2002)*, pp. 217–224. Kobe University, Japan (2002)
2. Berbner, R., Spahn, M., Repp, N., Heckmann, O., Steinmetz, R.: WSQoSX – A QoS architecture for Web Service workflows. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) *IC-SOC 2007*. LNCS, vol. 4749, Springer, Heidelberg (2007)
3. Solanki, M., Cau, A., Zedan, H.: Augmenting Semantic Web Service Description with Compositional Specification. In: *Proceedings of the WWW 2004*, vol. 1, pp. 544–552. ACM Press, New York (2004)
4. Mika, P., Oberle, D., Gangemi, A., Sabou, M.: Foundations for Service Ontologies: Aligning OWL-S to DOLCE. In: *Proceedings of the WWW 2004*, vol. 1, pp. 563–572. ACM Press, New York (2004)

5. Scholz, T., Timm, I.J., Spittel, R.: An Agent Architecture for Ensuring Quality of Service by Dynamic Capability Certification. In: Eymann, T., Klügl, F., Lamersdorf, W., Klusch, M., Huhns, M. (eds.) *MATES 2005. LNCS (LNAI)*, vol. 3550, pp. 130–140. Springer, Heidelberg (2005)
6. Liu, Y., Ngu, A.H.H., Zeng, L.: QoS Computation and Policing in Dynamic Web Service Selection. In: *Proc. of WWW 2004*, vol. 2, pp. 66–73. ACM Press, New York (2004)
7. Sheth, A., Cardoso, J., Miller, J., Kochut, K., Kang, M.: QoS for Serviceoriented Middleware. In: *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002)*, pp. 528–534 (2002)
8. Garvin, D.: *Managing Quality: The Strategic and Competitive Edge*. Free Press, NY (1988)
9. Ran, S.: A Model for Web Services Discovery With QoS. *SIGecom Exch. Journal* 4(1), 1–10 (2003)
10. Guttmann, C., Zukerman, I.: Towards Models of Incomplete and Uncertain Knowledge of Collaborators' Internal Resources. In: Lindemann, G., Denzinger, J., Timm, I.J., Unland, R. (eds.) *MATES 2004. LNCS (LNAI)*, vol. 3187, pp. 58–72. Springer, Heidelberg (2004)
11. Scholz, T., Timm, I.J., Woelk, P.-O.: Emerging Capabilities in Intelligent Agents for Flexible Production Control. In: Ueda, K., Monostri, L., Markus, A. (eds.) *Proceedings of the 5th Int. Workshop on Emergent Synthesis (IWES 2004)*, Budapest, pp. 99–105 (2004)
12. Riedemann, E.H.: *Testmethoden für sequentielle und nebenläufige Software-Systeme*. B.G. Teubner, Stuttgart (1987)
13. Menzies, T., Pecheur, C.: Verification and Validation and Artificial Intelligence. In: *Advances of Computers*, vol. 65, Academic Press, London (2005)
14. Booch, G.: *Objektorientierte Analyse und Design*. Addison-Wesley, Paris (1994)
15. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F.: *Lorensen: Object-Oriented Modelling and Design*. Prentice Hall, Englewood Cliffs, New Jersey (1991)
16. Hatton, L.: Does OO Sync. with How We Think? *IEEE Software* 15(3), 46–54 (1998)
17. Halladay, S., Wiebel, M.: *Object-oriented Software Engineering-Object Verification*. R&D Publications, Lawrence (1994)
18. Smith, M., Robson, D.: Object-Oriented Programming – The Problems of Validation. In: *Proc. of the Intl. Conf. on Software Maintenance*, IEEE Press, San Diego (1990)
19. Sneed, H.M., Winter, M.: *Testen objektorientierter Software*. München, Hanser
20. Timm, I.J., Scholz, T., Fürstenau, H.: From Testing to Theorem Proving. In: Kirn, S., Herzog, O., Lockemann, P., Spaniol, O. (eds.) *Multiagent Engineering – Theory and Applications in Enterprises*, pp. 531–554. Springer, Berlin (2006)
21. Hormann, A.: *Testbasierte Spezifikation von Agenteninteraktionsverhalten*. Diploma Thesis, University of Bremen (2006)
22. Liu, Y., Ngu, A.H.H., Zeng, L.: QoS Computation and Policing in Dynamic Web Service Selection. In: *Proceedings of the WWW 2004*, vol. 2, pp. 66–73. ACM Press, New York (2004)
23. Rovatsos, M., Schillo, M., Fischer, K., Weiß, G.: Indicators for Self-Diagnosis: Communication-Based Performance Measures. In: Schillo, M., Klusch, M., Müller, J., Tianfield, H. (eds.) *MATES 2003. LNCS (LNAI)*, vol. 2831, pp. 25–37. Springer, Heidelberg (2003)
24. Dash, R.K., Ramchurn, S.D., Jennings, N.R.: Trust-Based Mechanism Design. In: *Proceedings of the 3rd International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS 2004)*, pp. 748–755. ACM Press, New York (2004)
25. Falcone, R., Castelfranchi, C.: Trust Dynamics: How Trust is Influenced by Direct Experiences and by Trust itself. In: *Proc. of the 3rd Intl. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, pp. 740–747. ACM Press, New York (2004)
26. Woelk, P.-O., Rudzio, H., Zimmermann, R., Nimis, J.: Agent.Enterprise in a Nutshell. In: Kirn, S., Herzog, O., Lockemann, P., Spaniol, O. (eds.) *Multiagent Engineering – Theory and Applications in Enterprises*, pp. 73–90. Springer, Berlin (2006)

COMBEK 2008 PC Co-chairs' Message

COMBEK seeks to address the need for research that explores and embraces the novel, challenging but crucial issue of adapting knowledge resources to their user communities, and using them as a key instrument in building knowledge-intensive Internet systems. Through a deep understanding of the real-time, community-driven, evolution of shared knowledge structures and knowledge bases, including ontologies, folksonomies, and/or shared conceptual models, a Web-based knowledge-intensive system can be made operationally relevant and sustainable over long periods of time.

COMBEK's primary value proposition is to accentuate the role of community. The expected outcome is to innovate the science of ontology engineering and to unlock the expected (and unavoidable) paradigm shift in knowledge-based and community-driven systems. Such a paradigm would affect knowledge sharing and communication across diverse communities in business, industry, and society. We are further convinced that being a part of the On The Move Federated Conferences will turn a spotlight on the scientific issues addressed in COMBEK, making them visible and attractive to practitioners and industry.

COMBEK advocates transcending the current, narrow "ontology engineering" view on change management of knowledge structures that is at the heart of today's knowledge-intensive systems. We consider stakeholder communities as integral factors in the continuous evolution of the knowledge-intensive systems in which they collaborate. By bringing together researchers from different domains, COMBEK aims to advance research on a very broad spectrum of needs, opportunities, and solutions.

COMBEK offered a forum for practitioners and researchers to meet and exchange research and implementation ideas and results towards next-generation knowledge-intensive systems and radically new approaches in knowledge evolution. It gave practitioners and researchers an opportunity to present their work and to take part in open and extended discussions. Relevant topics included (but were not limited to) theoretical or empirical exploration and position papers on theory and method, as well as tool demonstrations, realistic case studies, and experience reports.

Out of 21 full-paper submissions, we selected 6 for inclusion in the official workshop proceedings, which results in an acceptance rate of less than 30%. We would like to thank all authors for their submissions and the members of the Program Committee for their time and rigor in reviewing the papers. The six papers address an interesting range of highly relevant research issues in this new field and will contribute to a better understanding of how the vision of community-based evolution of knowledge-intensive systems can be made a reality.

Van de Maele and Diaz introduce a two-phase approach to mapping heterogeneous semantic models, splitting up the traditional one-step process into a mapping phase that takes into account the actual community and context dimension, and a commitment phase. The proposal favors reuse, evolution, and scalability. Kotis motivates the needs for adopting semantic wiki technology for the delicate task argumentation and negotiation in collaborative ontology evolution. Debruyne et al. report on their work extending ontology negotiation with algorithms that afford (i) a statistical summary of

how community stakeholders modify and extend their shared ontology base; and (ii) a negotiation protocol for incorporating consensus changes to iterate the ontology.

Baatarjav et al. present a statistical technique for predicting the community an individual would belong to, based on Facebook social network data. This work leads to interesting insights for group recommendations in social network systems. Along this line, Phithakkitnukoon and Dantu apply machine learning techniques on call log data from the MIT Media Lab's reality mining corpus to classify mobile phone users into two social groups. The accuracy is enriched by feature selection based on a normalized mutual information heuristic. Finally, Van Damme and Christiaens introduce and demonstrate a novel approach to assess the semantic quality of tagging behavior in narrow folksonomies. In particular, they propose lexical ambiguity to feature the measure quality of tags.

This first workshop has helped us to establish a research community; at the same time, we are fully aware of the fact that most of the research work is still to be accomplished. We are looking forward to working on these challenges with all of our colleagues in this emerging research community.

November 2008

Pieter De Leenheer
Martin Hepp
Amit Sheth

Semi-automated Consensus Finding for Meaning Negotiation

Christophe Debruyne¹, Johannes Peeters², and Allal Zakaria Arrassi¹

¹ Semantics Technology and Applications Laboratory (STARLab)

Department of Computer Science

Vrije Universiteit Brussel

Pleinlaan 2, B-1050 BRUSSELS 5, Belgium

{chrdebru, aarrassi}@vub.ac.be

² Collibra nv

Brussels Business Base

Tweebeeck industrial park

rue de Ransbeek 230, B-1120 BRUSSELS, Belgium

johannes.peeters@collibra.com

Abstract. Finding a consensus between communities to create an ontology is a difficult task. An evolutionary process where domain experts and knowledge engineers work together intensively is needed to support collaborative communities in defining a common ontology. These communities model their view of a particular concept while knowledge engineers try to find a consensus. Negotiation, finding similarities and defining new points of interests are important processes to achieve such a consensus. To aid these processes we present several algorithms, built upon a state-of-the-art community grounded ontology evolution methodology. These algorithms are illustrated with an example.

1 Introduction

Communities share interests and information in order for them to collaborate. For such collaborations to be successful, they need instruments to achieve an agreement on how to communicate about those interests and information. One of these instruments is an ontology, which is a specification of a shared conceptualization [1]. The term quickly became popular in the field of knowledge engineering, where such conceptualizations are externalized formally in a computer resource. Externalization enables sharing and interoperation between information systems, the so called shared agreement [2].

When designing ontologies, we only want to conceptualize a relevant subset of the world as efficiently as possible [3]. An ontology engineering methodology is a vital instrument in order for collaborations to succeed. Such a methodology helps designing these formal, agreed and shared conceptualizations by making them reusable, reliable, shareable, portable and interoperable. It can also act as a basis for ontology engineering during class.

Starting from a common ontology, such a methodology requires the different communities to render their view of a new concept on the common ontology. This results in a divergence, which needs to be converged in order for a new version of the common ontology, containing an agreed view of that new concept, to appear. Ontology integration, of which a survey is provided in [4], takes care of that convergence. However, finding an agreement is a tedious task and requires all communities to work together and discuss their progress. These negotiation processes were discussed in approaches such as [5,6]. We name this process *meaning negotiation*.

2 Problem Statement

One of the biggest challenges of meaning negotiation, after receiving all the views, is to define ways to guide that process. Meaning negotiation can be a semi-automated process, where different algorithms help signalling problems or streamline the results for future iterations. When problems arise (e.g. a different semantics for a certain concept), the different communities need to be contacted for clarity, discussion or negotiation.

Questions can be formulated to indicate where problems might rise. The knowledge engineer can answer these questions in order to construct an agenda for the meaning negotiation session. The questions that need to be answered are threefold:

1. Which relations did the different communities specialize or generalize, by using more specialized or general concepts from a taxonomy?
2. Which are the stable parts in the ontology, type hierarchy or template? A stable part is a subset of concepts and their interrelationships, which were not changed by the different communities.
3. Which new concepts have to be elaborated in more detail, by defining relations around them, in a future iteration?

In this paper we propose a solution which answers the three questions mentioned above. We formulate the solution and present results from a realistic example in the following sections. We conclude with our findings and a discussion about some future directions, which we find worthwhile investigating.

3 Approach

In this section we present several algorithms answering the questions formulated in previous section. These algorithms are applied to the different views before the meaning negotiation session starts. The collaborative ontology engineering methodology that we extended is the DOGMA-MESS [3] ontology engineering methodology. We choose DOGMA-MESS for its explicit use of templates, which guide domain experts to give their conceptualization, not found in other ontology engineering methodologies such as HCOME [5] and DELIGENT [7].

3.1 DOGMA and DOGMA-MESS

The DOGMA¹ approach developed at STARLab aims at the development of a useful and scalable ontology engineering approach. In the DOGMA ontology engineering approach, ontology construction starts from a (possibly very large) uninterpreted base of elementary fact types called lexons [8,9] that are mined from linguistic descriptions such as existing schemas, a text corpus or formulated sentences by domain experts.

A lexon is an ordered 5-tuple of the form $\langle \gamma, t_1, r_1, r_2, t_2 \rangle$, where: t_1 and t_2 are respectively the *head-term* and *tail-term*, r_1 and r_2 the role and co-role and γ is the *context* in which the lexon holds. A lexon can be read in both directions since the head-term plays the role in the relation and the tail-term the co-role. The context is used to disambiguate the terms in case of homonyms, e.g.: a capital in a geographical context is not the same as a capital in an economical context. An example of a lexon is depicted in Fig. 1.

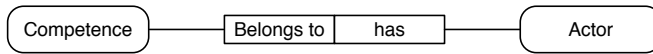


Fig. 1. Example of a lexon in some context γ

An ontological commitment to such a “lexon base” means selecting or reusing a meaningful set of facts from it that approximate the intended conceptualization, followed by the addition of a set of constraints, or rules, to this subset. This commitment process is inspired by the fact-based database modeling method NIAM/ORM [10].

DOGMA was extended with a “Meaning Evolution Support System”, resulting in an interorganizational ontology engineering methodology called DOGMA-MESS [3]. The common ontology is produced in a number of iterations, where each iteration consists of four stages [11]:

1. *Community Grounding*: The core domain expert and the knowledge engineer identify the relevant key concepts with their relations and externalize them in an ontology. This ontology contains conceptualizations common to and accepted by the community.
2. *Perspective Rendering*: In this phase, each stakeholder’s domain expert renders its perspective on the common ontology, by specializing it to their idea.
3. *Perspective Alignment*: In perspective alignment a new proposal for the next version of the common ontology is produced. Relevant material from both the common ontology and the different perspectives is gathered and aligned. The methodology allows domain experts to render their view not only by specialization, which allows new definitions to be created, but also by generalization. A consequence of this “creative chaos” is that the alignment process is far from trivial. During that process all the domain and core domain experts need to collaborate.

¹ DOGMA: Developing Ontology-Grounded Methods and Applications.

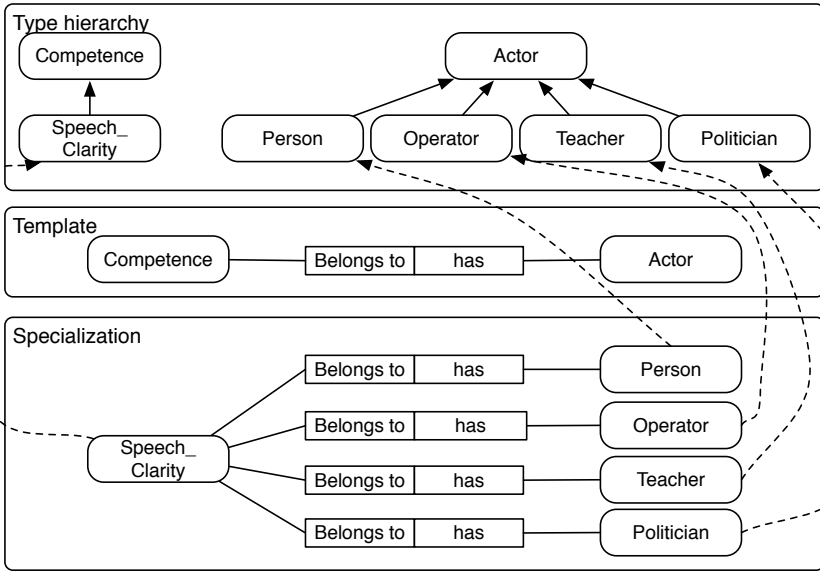


Fig. 2. Example of how an organization or community can specialize a given template using concepts from a taxonomy

4. *Perspective Commitment:* In this phase the part of the ontology that is aligned by the community forms the next version of the common ontology. All participants finally commit their instance bases (e.g. databases) and applications to the new version of the ontology.

A *template* is provided to the different communities in order to model the relevant parts. This template can be described as a generalization of a pattern, either found in the models or through experience of the knowledge engineers. An example of such a template is given in Fig. 2. Given a template and a taxonomy, domain experts can *specialize* the relations in the template’s by replacing the concepts in that relation by one of its children in the taxonomy. Domain experts can also extend the taxonomy by introducing new concepts. The latter is also called a *specialization* of a template. A revision of the template is needed when domain experts introduce new relations rather than specialize the given set of relations, thus making templates also subject to change and evolution.

3.2 Algorithms and Experiment

In this section we will present the algorithms. The algorithms are illustrated with a running example, based on an experiment held at STARLab. Their combined output gives means to answer the three questions formulated in Section 2. Note that there is no one-to-one mapping between algorithms and questions.

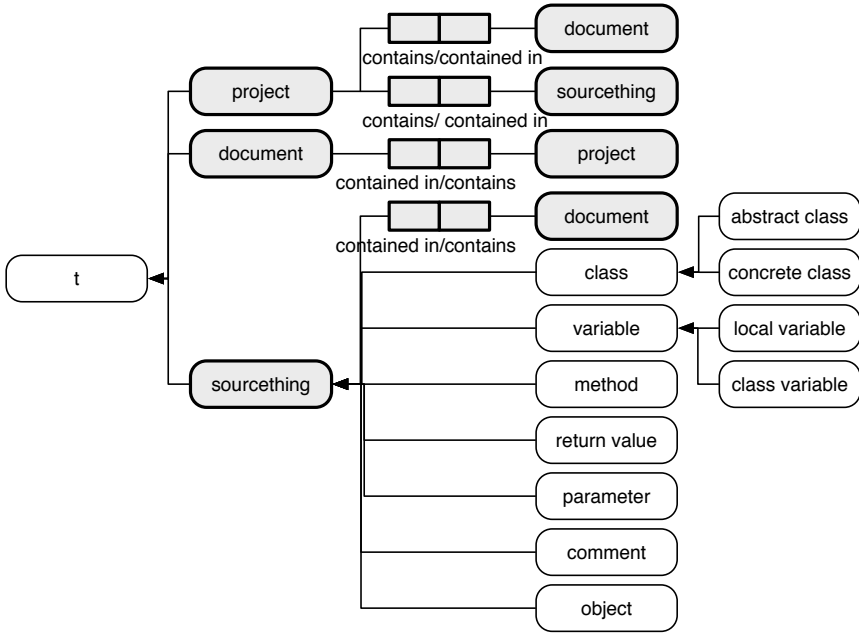


Fig. 3. First version of the template based on the first organization’s model. The arrows denote the type hierarchy. The terms and role which are colored denote the binary relations that make up the template.

Experimental Setup. The experiment involved five groups of students representing fictive organizations with different core businesses: bug tracking, versioning, document annotation, source code annotation and module management. The goal was to model a part of the open source community. The first group provided the basis of a first version of a template, which is depicted in Fig. 3.

The template was given to the remaining four organizations, which had to specialize it. These specializations had to be examined for differences, similarities and conflicts.

Files and images concerning the template, specializations and descriptions of each of the organization’s core businesses as well as the output of the algorithms can be found at <http://wilma.vub.ac.be/~chrdebru/combek/index.html>

Algorithm 1. This algorithm looks at the different specializations to determine which relations of the template were specialized or generalized. We first define a function $relationsOf : C \times T \rightarrow L$, where C is the set of concepts, T is the set of templates and L is the set of all possible sets of lexons. The function $relationsOf$ returns a subset of lexons in a template $t \in T$ where a concept $c \in C$ either plays the role or the co-role.

Given a template $t \in T$ and a set of specializations $S = \{s_1, \dots, s_n\}$ by the different communities, the algorithm works as follows:

- For each concept $c \in C$ in t playing either the role or co-role in a binary relation that needs to be specialized:
 - $rels \leftarrow relationsOf(c, t)$
 - For each s_i , where $i = 1 \rightarrow |S|$
 - * if $c \in s_i$ then
 - Look at the taxonomic children of c in s_i , either from the common ontology or introduced by the organization. If those children were used in a specialization of one of the relations in $rels$, we have found a specialization.
 - Look at the taxonomic parents of c in s_i . If the concept c in s_i has relations not appearing in $rels$, but appearing in one of its taxonomic parents² and that same relation does not appear in T , we have found a generalization.

This algorithm, which returns the specializations and generalizations by the organizations, is straightforward and answers the first of the three questions.

Results of Algorithm 1. Only one organization enriched the taxonomy under one of the existing terms (see Fig. 4). However, none of the organizations had actually specialized any of the provided binary relations. This was mainly due to the nature of the experiment, where there was no clear ‘common goal’ for each of the stakeholders. The organizations were asked to model a part of the open source community instead, which they did by introducing new concepts and relations needed for their own core business.

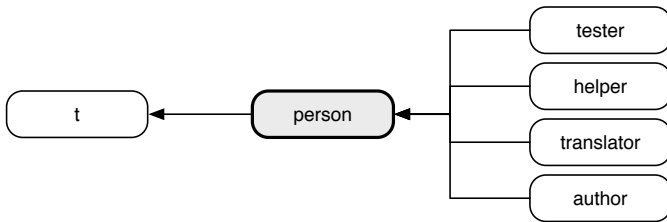


Fig. 4. Some of the terms introduced under the concept ‘person’ by one of the organizations

Algorithm 2. The second algorithm will look for candidate concepts in the different views, which might get introduced in a new version of the common ontology. The second algorithm also provides an answer to the question of which are the stable parts in the ontology.

For this process we define a function $poc : C \times O \rightarrow C$, where C is the set of all concepts, O the set of all ontologies, templates and specializations. The function searches a concept in an ontology $o \in O$ with a certain label and

² The taxonomic parents in the specializations can differ because organizations are allowed to change the concept’s place in the type hierarchy.

returns its parent in the type hierarchy, *poc* stands thus for *Path of Concept*. If the ontology does not contain such a concept, then the functions returns nothing.

We construct a matrix where we create a row for each concept $c \in C$ and a column for the template and each specialization. Then in each cell of the matrix we output the result of the function *poc* for that particular concept and template/specialization.

Concepts	Template T	Specilization S_1	...	Specialization S_n
$label_1$	$poc(label_1, T)$	$poc(label_1, S_1)$...	$poc(label_1, S_2)$
$label_2$	$poc(label_2, T)$	$poc(label_2, S_1)$...	$poc(label_2, S_2)$
$label_3$	$poc(label_3, T)$	$poc(label_3, S_1)$...	$poc(label_3, S_2)$
...

We can determine what an organization did for a certain concept: introduction, removal (if the concept does not appear in the column of the organization, but does appear in the column of the template) or nothing at all.

We can now take possible actions for the concepts given a certain threshold ϵ . We assign each concept a score, which is the number of times the concept appears in the template or one of the specializations divided by the number of specializations + 1 (for the template). If the majority of the organizations removed a certain concept, the concept's score will be lower than ϵ and hence become a candidate for dropping. If a certain number of organizations introduced a concept (not considering the taxonomy), the score goes up. When the score of a concept is greater than or equal to ϵ , the concept becomes a candidate for introducing.

As already mentioned, this algorithm answers two of the three questions, namely which concepts are considered interesting for a next iteration and which are the stable parts in the specializations. The latter is answered by looking at the results of the *poc* function. If most (if not all) concepts have the same parent, we consider them stable.

Results of Algorithm 2. The second algorithm showed that, given a certain threshold, both *version* and *person* could be added to a next version of the ontology. Both terms are also candidates for future iterations of the process. The resulting matrix (see Table 1 for a part of the matrix) of the second algorithm also showed that most concepts in the template are stable. One organization restructured the template's taxonomy; making only everything below the concept *sourcething* truly stable. Note that concepts such as *account* and *admin*, which do not occur in the template found in Fig. 3, have been introduced by one or more organizations.

When interviewing the different groups about the taxonomy; the groups that did not alter the type hierarchy argued that such operations would have cost too much if commitments to the ontology were already made. The group that

Table 1. Part of the matrix constructed by the second algorithm

	Template	Org ₁	Org ₂	Org ₃	Org ₄	Score
_Object					T	0.2
Abstract class	Class	Class	Class	Class	Class	1.0
Account			Person			0.2
Admin			Person			0.2
Author					Person	0.2

changed the type hierarchy found that such (possibly costly) operations in early stages of ontology engineering are ground, provided if they would benefit the reusability and correct level of abstraction of the ontology in the long run.

Algorithm 3. The third and last algorithm will look for candidate relations between concepts, which will be introduced in a next version of the common ontology. This algorithm uses the output of the second algorithm.

The algorithm works as follows: given a second threshold $\alpha \leq \epsilon$, for each concept c marked as a candidate for introduction in the previous algorithm:

- List all the relations of that concept for each specialization.
- Register the occurrences of each of these relations across all specializations, solely looking at the lexical representation of these relations (or lexons).
- If the average occurrence of a relation is greater than or equal to α , the relation and (if not yet introduced) the other concept, are included in a next version of the template as well.

Results of Algorithm 3. Two organizations introduced the term *version* and *person* was introduced by three organizations. Due to the very different natures of the core businesses; the organizations have not introduced similar relations between concepts.

3.3 Meaning Negotiation

The results of these three algorithms were taken to the meaning negotiation session, where they were discussed. During this discussion some problems arose, which became points of the agenda of that meeting. One of these problems was the difference between the terms *version* and *revision*, since the results marked the term *version* as a candidate for introduction and not *revision*. Some organizations argued they denote the same concept. After discussing the context of both terms (and the relations around the concepts), all organizations agreed they were different.

One organization restructured the type hierarchy to benefit future reuse, while others considered this too much of a risk (and a cost) when commitments to that ontology were already made. Since the majority of the stakeholders did not want to risk that cost, the changes to the type hierarchy were ignored. We also found

that providing a very basic taxonomy would have encouraged the organizations to introduce new terms more neatly in a type hierarchy.

4 Discussion

One of the most difficult tasks in ontology engineering is to find a consensus between different stakeholders while trying to define a common ontology. Meaning negotiation, where problems and conflicts within the different views among the stakeholders are discussed, is therefore an important process. This process can be guided by answering three questions about the different views: (1) what relations have been generalized or specialized, (2) what are the stable parts in the ontology and (3) what are new interesting concepts that have to be modelled in future iterations. These three questions will point out the problems, which can be used to construct an agenda for the meaning negotiation process.

In this paper we extended a state of the art collaborative ontology engineering methodology, called DOGMA-MESS, with three algorithms answering these proposed questions. These algorithms were applied in an experiment where groups of students simulated the methodology to model a part of the open source community by representing fictive organizations. The list of similarities and conflicts between the different models was limited due to the very different core business of the organizations, but the algorithms showed their potential in constructing an agenda for a meaning negotiation session.

5 Future Work

The algorithms were currently applied on the models at lexical level, applying them while also taking into account the type hierarchy or glossaries would add semantics to these algorithms.

Even though the results proved promising, the constructed algorithms still need to be validated with a use case of substantial size. Such an experimental setup will also result in an implementation of the algorithms.

Another trail worth investigating is applying the extension to other collaborative ontology engineering methodologies like HCOME [\[5\]](#) and DELIGENT [\[7\]](#). The difference between those methodologies and DOGMA-MESS is the explicit use of templates, which is used as input by the algorithms, by the latter.

Acknowledgements

We would like to thank our Professor Robert Meersman and everyone at VUB STARLab for aiding us with this experiment as well as proposing us to try and publish the results. We would also like to thank Simon Janssens, a Computer Science student at the VUB, for assisting us with this experiment.

References

1. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 5, 199–220 (1993)
2. Jarrar, M., Meersman, R.: 3. In: *Ontology Engineering - The DOGMA Approach. Advances in Web Semantic, A state-of-the Art Semantic Web Advances in Web Semantics IFIP2.12*, vol. 1, Springer-sbm, Heidelberg (2007)
3. de Moor, A., De Leenheer, P., Meersman, R.: DOGMA-MESS: A meaning evolution support system for interorganizational ontology engineering. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) *ICCS 2006. LNCS (LNAI)*, vol. 4068, pp. 189–202. Springer, Heidelberg (2006)
4. Kalfoglou, Y., Schorlemmer, W.M.: Ontology mapping: The state of the art. In: Kalfoglou, Y., Schorlemmer, W.M., Sheth, A.P., Staab, S., Uschold, M. (eds.) *Semantic Interoperability and Integration. Dagstuhl Seminar Proceedings, Dagstuhl, Germany, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI)*, vol. 04391 (2005)
5. Kotis, K., Vouros, A.: Human-centered ontology engineering: The hcome methodology. *Knowledge and Information Systems* 10, 109–131 (2006)
6. Kunz, W., Rittel, H.: Issues as elements of information systems. Working Paper 131, Institute of Urban and Regional Development, University of California, Berkeley, California (1970)
7. Pinto, H.S., Staab, S., Tempich, C.: Diligent: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In: *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI)*, pp. 393–397. IOS Press, Amsterdam (2004)
8. Meersman, R.: Semantic ontology tools in is design. In: Raś, Z.W., Skowron, A. (eds.) *ISMIS 1999. LNCS*, vol. 1609, pp. 30–45. Springer, Heidelberg (1999)
9. Meersman, R.: Reusing certain database design principles, methods and design techniques for ontology theory, construction and methodology. Technical report, VUB STARLab (2001)
10. Halpin, T.A.: *Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design*. Morgan Kaufmann Publishers Inc., San Francisco (2001)
11. De Leenheer, P.: Towards an ontological foundation for evolving agent communities (2008); Invited paper for IEEE ESAS 2008

On Supporting HCOME-3O Ontology Argumentation Using Semantic Wiki Technology (Position Paper)

Konstantinos Kotis

University of the Aegean, Dept. of Information & Communications Systems Engineering,
AI Lab, 83200 Karlovassi, Greece
kotis@aegean.gr,
<http://www.icsd.aegean.gr/ai-lab>

Abstract. To support the sharing of consistently evolved and living ontologies within and across different communities, HCOME-3O framework has been recently proposed. The framework introduces a set of three (meta-) ontologies for capturing the meta-information that is necessary for interlinking, sharing, and combining knowledge among the parties involved in a collaborative (domain) ontology engineering process. Although a prototype software (namely HCONE) based on this framework has been developed, collaborative tasks embedded in the HCOME methodology such as the *ontology argumentation* could be alternatively designed using *open* and *Web community-driven* (collective intelligence-based) technologies. In this short paper we state our position that the existing technology used to develop a Semantic Wiki (and its extensions) can be re-used in HCOME-3O-based tools in order to support Web community-driven collaborative ontology engineering tasks.

Keywords: ontology argumentation, collaborative ontology engineering, Semantic Wiki, HCOME methodology, collective intelligence.

1 Introduction

Ontologies are *evolving* and *shared* artefacts that are collaboratively and iteratively developed, evolved, evaluated and discussed within communities of knowledge workers. To enhance the potential of ontologies to be collaboratively engineered within and between different communities, these artefacts must be escorted with *all* the necessary information (namely meta-information) concerning the conceptualization they realize, implementation decisions and their evolution.

In HCOME-3O framework [1], authors proposed the integration of three (meta-) ontologies that provide information concerning the conceptualization and the development of domain ontologies, the atomic changes made by knowledge workers, the long-term evolutions and argumentations behind decisions taken during the lifecycle of an ontology. Figure 1 depicts ontology engineering tasks for a *domain* ontology and its versions, i.e. editing, argumentation, exploiting and inspecting, during which meta-information is captured and recorded (in *development* ontologies) either as information concerning a simple task or as information concerning the interlinking of

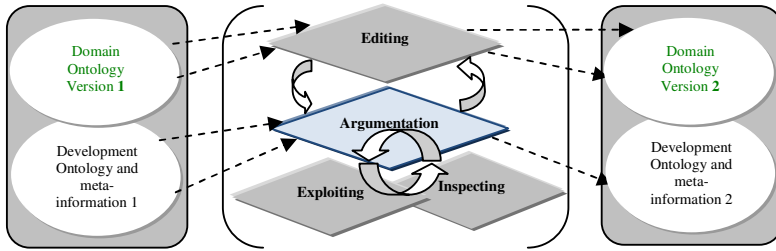


Fig. 1. The HCOME-3O framework for recording interlinked meta-information concerning ontology engineering tasks

tasks. This framework has been proposed in the context of HCOME collaborative engineering methodology [2]. HCOME places major emphasis on the conversational development, evaluation and evolution of ontologies, which implies the extended sharing of the constructed domain ontologies together with the meta-information that would support the interlinking, combination, and communication of knowledge shaped through practice and interaction among community members.

More specifically, in the context of HCOME-3O framework, ontology argumentation task is supported by the “Argumentation ontology”¹. Such ontology provides a schema for representing meta-information about *issues*, *positions*, and *arguments* that contributing parties make during an argumentation dialogue upon the collaborative evolution of shared ontologies. Specifically, an argument may raise an issue that either suggests changes in the domain conceptualization, or questions the implementation of the conceptualized entities/properties. Based on this issue, a collaborative party may respond by publicizing a position, i.e. a new version of the ontology, or by suggesting the change of a specific ontology element. A new argument may be placed for or against a position, and so on. Issues may be generalized or specialized by other issues. The connection of the recorded arguments with the ontology elements discussed by specific contributing parties and with the changes made during a period is performed through the argumentation item and position classes’ properties (formal item, contributing party, period and evolving ontology). The argumentation ontology supports the capturing of the structure of the entire argumentation dialogue as it evolves among collaborating parties within a period. It allows the tracking and the rationale behind atomic changes and/or ontology versions. It is generic and simple enough so as to support argumentation on the conceptual and on the formal aspects of an ontology.

Current implementation of the ontology argumentation functionality of HCOME-3O framework is captured in HCONE prototype tool [1], based on a stand-alone JAVA implementation (front-end) and JENA support for ontology management (persistent storage). Although it is a platform in-dependent implementation, the functionality is not open to the Web community. The use of open, Web community-driven technology, such as Wiki technology, in order to enable collaborative and

¹ An OWL implementation of the argumentation ontology can be accessed from <http://www.icsd.aegean.gr/ai-ab/projects/HCONEv2/ontologies/HCONEarguOnto.owl>

community-driven ontology engineering (by giving users with no or little expertise in ontology engineering the opportunity to contribute) is not new. In order to support open and Web community-driven ontology argumentation, existing Wiki technology can be integrated.

For designing and developing an open and Web community-driven ontology argumentation functionality embedded in a collaborative ontology engineering environment that is based on HCOME-3O framework, the following requirements should be met:

1. Use an Argumentation Ontology to represent meta-information concerning the recording and tracking of the structured conversations. Record such meta-information as individual elements (instances) of the Argumentation Ontology classes. Any Argumentation Ontology can be used, given that it will be interlinked with the HCOME-3O (meta-) ontologies.
2. Use the HCOME-3O (meta-) ontology framework to record meta-information concerning the interlinking between conversations and ontology development and evolution (changes and versions of a domain ontology). The recording of interlinking between (meta-) ontologies is what really supports the sharing of consistently evolved and living ontologies within and across different communities [1].
3. Use of Semantic Wiki technology for openness and Web community-driven engineering. Developing collaborative functionalities of ontology engineering, such as ontology argumentation, is much more easy and efficient when we use technologies that were devised for such purposes.

The aim of this paper is to state author's position concerning the use of (semantic) Wiki technology for supporting ontology argumentation task in O.E tools that have been (or going to be) designed according to HCOME-3O framework.

2 Related Work and Motivation

In myOntology project [3] the challenges of collaborative, community-driven, and wiki-based ontology engineering are investigated. The simplicity of Wiki technology and consensus finding support by exploiting the collective intelligence of a community is being used to collaboratively develop lightweight ontologies. myOntology goal is not only to allow co-existence and interoperability of conflicting views but more importantly support the community in achieving consensus similar to Wikipedia, where one can observe that the process of consensus finding is supported by functionality allowing discussion (however, not structured dialogues).

In NeOn project, the Cicero web-based tool [4] supports asynchronous discussions between several participants. This social software application is based on the idea of Issue Based Information Systems (IBIS) and the DILIGENT argumentation framework [5]. The DILIGENT argumentation framework was adapted for Cicero in order to make it easier applicable on discussions and in order to reduce the learning effort by users. In Cicero, a discussion starts with defining the issue which should be discussed. Then possible solutions can be proposed. Subsequently, the solution proposals are discussed with the help of supporting or objecting arguments.

Both works provide strong evidences that collective intelligence in the form of (semantic) Wikis can be used to support collaborative ontology engineering, with the advantages of openness and scalability. As far as concerns reaching a consensus on a shared ontology during argumentation, both works, although they provide mechanisms to record the actual dialogues, meta-information concerning the recording of the interlinking between conversations and ontology evolution (versions of a domain ontology) is not recorded.

Our position statement in this paper has been motivated by these related technologies. We conjecture that the related technologies must aim to their integration with the HCOME-3O framework since the recording of interlinking between (meta-) ontologies is what really supports the sharing of consistently evolved and living ontologies within and across different communities [1]. By re-using such technologies and extending them to be compliant with HCOME-3O framework it is possible to achieve this goal.

3 Wiki-Based HCOME-3O Ontology Argumentation

Following the HCONE tool design requirements as these were implied by HCOME-3O framework [1], we introduce a personal and a shared space for performing ontology engineering tasks. In this paper an initial architecture for the design of a HCOME-3O-based ontology engineering tool that integrates Semantic Wiki technology (currently for the ontology argumentation task only) is proposed (Figure 2). The proposed architecture, following the “Exploitation” phase of HCOME methodology, supports the following tasks:

1. The inspecting of shared ontologies (reviewing, evaluating and criticizing specified conceptualizations),
2. The inspecting (comparison) of shared versions of an ontology, for identifying the differences (tracking changes) between them,
3. The posting of arguments upon versions of ontologies for supporting decisions for or against specifications.

Although tasks 1 and 2 can be performed in the personal space, it has been already shown in other lines of HCONE research [2] that they could also be performed collaboratively in the shared space. Given that existing technology can support it, the *Exploiting* and *Inspecting* tasks could be performed to the shared space using extensions of Semantic Wiki technology such as the Halo² extension. Allowing however the execution of these tasks in both spaces may be a “gold” design solution (must be evaluated with further work). The *Editing* task can also be moved to the shared space, since technologies have been already proposed that can support it [3]. However, only the editing of lightweight ontologies can be (with existing and proposed technologies) supported. Finally, the *Argumentation* task can be executed in the shared space since technology is mature enough to support it in an open and Web community-driven environment [4].

² http://semanticweb.org/wiki/Halo_Extension

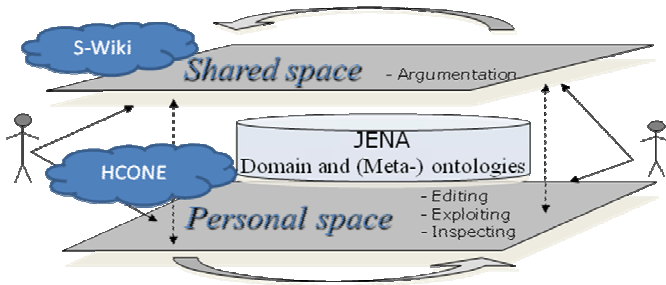


Fig. 2. An HCOME-3O-based ontology engineering architecture integrating Semantic Wiki technology

To meet the requirements outlined in the Introduction section, the above presented ontology engineering tasks should be integrated in environments that have been designed according to HCOME-3O framework for recording meta-information and their interlinking. Concerning the Ontology Argumentation task, integrating a Wiki-based tool such as CICERO with stand-alone Ontology Engineering tools (e.g. HCONE [1]) that store meta-information in JENA ontologies, is feasible. The integration of such tools requires their extension in order to communicate information concerning ontology argumentation (at the Ontology Argumentation Wiki side e.g. CICERO) and ontology evolution (at the Editing tool side e.g. HCONE) meta-information.

In order to test the proposed in this paper technology, we have developed an experimental Ontology Argumentation Wiki (namely, HCOMEasWiki) using the freely available PHP API's of MediaWiki³, Semantic MediaWiki⁴, and CICERO (see Figure 3). As depicted, a first "Issue" has been created for discussion under the title "My first issue", concerning the specification of an ontology class ("It concerns the ontology class..."). There are no "Reactions" (Arguments or Solution Proposals) for this "Issue", according to CICERO technology.

To integrate CICERO Ontology Argumentation functionality in HCOME-3O framework-based HCONE tool, some variables (corresponding to properties or classes of the ontologies from both tools) should be mapped and some others to be introduced, as shown in the Table 1. The table is not complete, but it gives an idea of what is needed to be done at the design level in order to easily extend HCONE tool's ontology argumentation functionality with the Semantic Wiki technology. Having said that, it must be also stated that CICERO is being used for evaluating the prototype version of the proposed approach. Several limitations of CICERO impose several new variables (property or category type of Wiki pages) that need to be introduced prior integrating it with HCONE tool. However, other similar implementations designed in accordance to the Ontology Argumentation framework proposed in HCOME [1, 2] could be more easily integrated in the proposed approach.

³ www.mediawiki.org/

⁴ http://semanticweb.org/wiki/Semantic_MediaWiki

Table 1. Example integration actions towards a Wiki-based HCOME-3O ontology argumentation

HCOMEasWiki wiki	Integration Action	HCONE tool
“Created by” property (e.g. “Kotis”)	Map	“Name” and “Surname” variables corresponding to properties of Contributing_Party class of Administrative (meta-) Ontology
“Discussed_Element_Class” “Discussed_Element_Individual” “Discussed_Element_Property”	Introduce categories in Wiki and then Map	“Class”, “Individual”, “Property” variables corresponding to sub-classes of “Element” class of Administrative (meta-) Ontology
“Discussed_Ontology”	Introduce categories in Wiki and then Map	“Evolving Ontology” variable corresponding to sub-class of “Ontology” class of Administrative (meta-) Ontology
“Issue” Category (e.g. “My first issue”)	Map	“Issue” variable corresponding to “Issue” class of Argumentation (meta-) Ontology

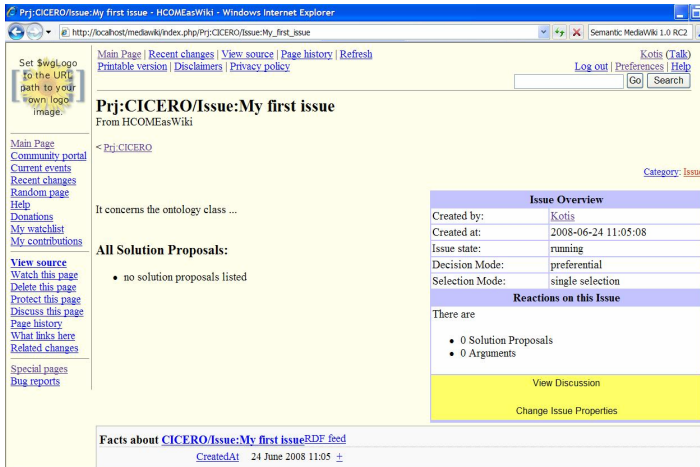


Fig. 3. Preliminary experimentations with Ontology Argumentation Wiki’s (CICERO snapshot)

4 Conclusions

The aim of this paper is to state author’s position concerning the use of (semantic) Wiki technology for supporting ontology argumentation task in O.E tools that have been (or going to be) designed according to HCOME-3O framework. An initial architecture for the design of a HCOME-3O-based ontology engineering tool that integrates Semantic Wiki technology (currently for the ontology argumentation task only) is proposed and preliminary experimentation with such technology is reported.

Apart from completing the proposed approach and experimentation towards reporting pros and cons from its full scale evaluation, future work includes further consideration of important methodological implications: Wiki-based ontology engineering approaches (and Wiki-based ontology argumentation consequently) are based on a self-organization principle, where concepts emerge uncontrolled following from the implicit community interests (bottom-up approach). On the other hand, middle-out approaches [2], [5] and [6] provide knowledge workers with a clear direction/focus to render their perspectives on an initial common ontology (which was grounded using a top-down approach). This focus is translated in concept-templates that are most urgent to be filled and are derived from the community discussion (socialization). Top-down (traditional approaches that are based on the knowledge engineer), bottom-up, and middle-out ontology engineering approaches should be seen as complementary. A key challenge is to find a balance between them, a point that will be accepted by knowledge workers during their ontology engineering practice [6].

Acknowledgments. Author thanks reviewers for their comments and acknowledges valuable contribution of the research teams that have worked on HCOME and DOGMA-MESS ontology engineering approaches.

References

1. Vouros, G.A., Kotis, K., Chalkiopoulos, C., Lelli, N.: The HCOME-3O Framework for Supporting the Collaborative Engineering of Evolving Ontologies. In: ESOE 2007 International Workshop on Emergent Semantics and Ontology Evolution, ISWC 2007, Busan, Korea, CEUR-WS.org, November 12, 2007, vol. 292 (2007)
2. Kotis, K., Vouros, G.A.: Human-Centered Ontology Engineering: the HCOME Methodology. *International Journal of Knowledge and Information Systems (KAIS)* 10(1), 109–131 (2006)
3. Siorpaes, K., Hepp, M.: MyOntology: The Marriage of Collective Intelligence and Ontology Engineering. In: *Proceedings of the Workshop Bridging the Gap between Semantic Web and Web 2.0, ESWC (2007)*
4. Dellschaft, Klaas, Engelbrecht, Hendrik, MonteBarreto, José, Rutenbeck, Sascha, Staab, S.: Cicero: Tracking Design Rationale in Collaborative Ontology Engineering. In: *Proceedings of the ESWC 2008 Demo Session (2008)*
5. Pinto, H.S., Staab, S., Sure, Y., Tempich, C.: OntoEdit empowering SWAP: a case study in supporting Distributed, Loosely-controlled and evolvinG Engineering of oNTologies (DILIGENT). In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) *ESWS 2004. LNCS*, vol. 3053, Springer, Heidelberg (2004)
6. de Moor, A., De Leenheer, P., Meersman, R.: DOGMA-MESS: A Meaning Evolution Support System for Interorganizational Ontology Engineering. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) *ICCS 2006. LNCS (LNAI)*, vol. 4068, pp. 189–203. Springer, Heidelberg (2006)

Inferring Social Groups Using Call Logs

Santi Phithakkitnukoon and Ram Dantu

Department of Computer Science and Engineering
University of North Texas, Denton, TX 76203, USA
{santi, rdantu}@unt.edu

Abstract. Recent increase in population of mobile phone users makes it a valuable source of information for social network analysis. For a given call log, how much can we tell about the person's social group? Unnoticeably, phone user's calling personality and habit has been concealed in the call logs from which we believe that it can be extracted to infer its user's social group information. In this paper, we present an end-to-end system for inferring social networks based on "only" call logs using kernel-based naïve Bayesian learning. We also introduce normalized mutual information for feature selection process. Our model is evaluated with real-life call logs where it performs at high accuracy rate of 81.82%.

Keywords: Social groups, Call logs.

1 Introduction

Social network describes a social structure of social entities and the pattern of inter-relationships among them. A social network can be either face-to-face or virtual network in which people primarily interact via communication media such as letters, telephone, email, or Usenet. Knowledge of social networks can be useful in many applications. In commerce, viral marketing can exploit the relationship between existing and potential customers to increase sales of products and services. In law enforcement, criminal investigation concerning organized crimes such drugs and money laundering or terrorism can use the knowledge of how the perpetrators are connected to one another to assist the effort in disrupting a criminal act or identifying additional suspects.

Social computing has emerged recently as an exciting research area which aims to develop better social software to facilitate interaction and communication among groups of people, to computerize aspects of human society, and to forecast the effects of changing technologies and policies on social and cultural behavior. One of the major challenges in social computing is obtaining real-world data. Quite often, analysis is based on simulations.

With rapidly increasing number of mobile phone users, mobile social networks have gained interests from several research communities. We also find it interesting to study the relationships between mobile phone users' calling behaviors and their social groups. With availability of real-life data of mobile phone users' call logs collected by Reality Mining project group [1], it allows us to carry out our analysis

and experimental results in this paper where we propose an end-to-end system for inferring social groups based solely on call logs. We believe that phone user’s calling personality and habit has been unnoticeably concealed in the call logs from which it can be extracted to infer its user’s social group information. To the best of our knowledge, no scientific research has been reported in classifying social networks/groups based solely on call logs.

The rest of the paper is organized as follows. In section 2, the system overview is presented. Section 3 describes our real-life dataset collected from mobile phone users. Data extraction process is then carried out in section 4 with preliminary statistical analysis. Section 5 discusses feature selection process and introduces normalized mutual information for selecting useful features. Kernel-based naïve Bayesian classifier is presented in section 6. The performance evaluation of proposed system is carried out through implementation in section 7. Finally, in section 8, we summarize our findings and conclude this paper with an outlook on future work.

2 System Overview

The system described here is intended to perform social group classification based on personal phone records. The input is phone records or call logs showing pertinent information (number dialed, duration, time of communications, etc.). The call log are then transformed into knowledge useful for the classifier by extracting calling patterns and selecting useful features. The kernel-based naïve Bayesian classifier is used to perform supervised classification based on computed probability using kernel density estimator.

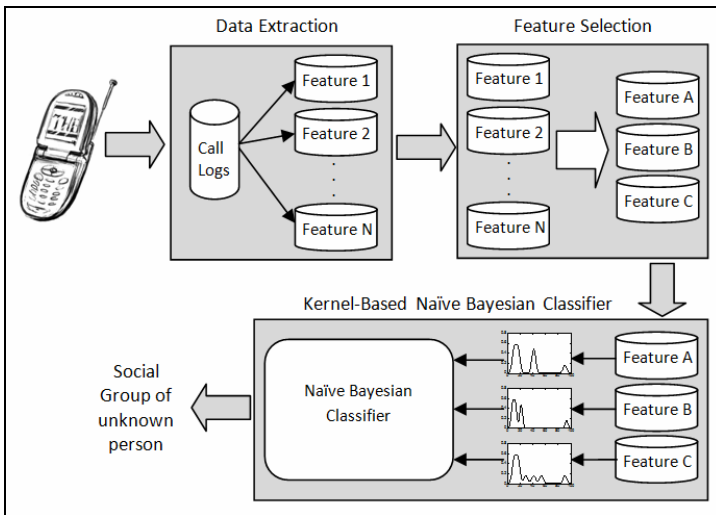


Fig. 1. System overview

3 Real-Life Dataset

Every day phone calls on the cellular network include calls from/to different sections of our social life. We receive/make calls from/to family members, friends, supervisors, neighbors, and strangers. Every person exhibits a unique traffic pattern. Unnoticeably, phone user's calling personality and habit has been concealed in the call logs from which we believe that it can be extracted to infer its user's social networks information. To study this, we use the real-life call logs of 94 individual mobile phone users over the course of nine months which were collected at Massachusetts Institute of Technology (MIT) by the Reality Mining project group [1]. Call logs were collected using Nokia 6600 smart phones loaded with software written both at MIT and the University of Helsinki to record minutely phone information including call logs, users in proximity, locations, and phone application currently being used. Of 94 phone users, 25 were incoming Sloan business students while the remaining 69 users were associated with the MIT Media Lab. According to MIT [1], this study represents the largest mobile phone study ever in academia and the data collected can be used in a variety of fields ranging from psychology to machine learning. There are some research works conducted by the Reality Mining project group using this dataset involves relationship inference, user behavior prediction, and organizational group dynamics.

As previously mentioned, our interest and the focus of this paper is to extract the phone user's behavior concealed in the call logs and attempt to accurately classify user into belonging social networks. With MIT dataset, classification can be performed to differentiate phone users from the Media Lab and from Sloan. As described earlier, MIT dataset consists of more than just call logs but location information and others. In order to be more generalized, only call logs are considered for our study as currently call logs are only accessible feature from service providers (e.g. billing, online account).

Due to missing information on the dataset which leaves us 84 users instead of 94 users, we then have 22 Sloan users and 62 Media Lab users. Of 62 Media Lab users, 20 users are clearly marked as students. We believe that even though all 62 users are with Media Lab, sub-social groups can be formed such as students, faculty, and staff, which exhibit slightly different calling behavior. Therefore we choose to perform classification between clearly marked Media Lab students and Sloan students.

4 Data Extraction

The main goal is to find some features from the call logs (raw data) that can solidly differentiate Media Lab students and Sloan students. For the data extraction process, we try to retrieve as much as possible useful features from the call logs. There might not be one dominate feature that captures entire calling behavior but combination of those characterize the core behavior structure. There are 11 features extracted and listed in Table 1 along with some statistical analysis (i.e. averages (Avg.) and standard deviations (Std.)) where feature descriptions are listed in Table 2.

From the first glance of these features and their statistics, it is clear that there are differences between two social networks however the differences are not adequately large enough to differentiate them based on each individual feature.

Table 1. Extracted features

Features	Media Lab		Sloan	
	Avg.	Std.	Avg.	Std.
All_calls	9.670	6.902	14.168	7.264
Inc_calls	2.920	2.542	3.756	2.197
Out_calls	6.750	4.501	10.413	5.549
Missed_calls	8.708	6.167	12.810	6.718
All_talk	246.716	304.899	196.966	260.884
Inc_talk	140.906	109.479	172.518	111.427
Out_talk	272.599	367.231	207.328	320.731
All_call_time	12.934	1.671	14.571	2.120
Inc_call_time	13.164	1.775	14.591	2.226
Out_call_time	12.881	1.752	14.583	2.121
Ent_call_time	6.137	0.683	4.059	0.553

Table 2. Extracted feature descriptions

Features	Feature description
All_calls	The total number of all calls per day including incoming, outgoing, and missed calls.
Inc_calls	The number of incoming calls per day.
Out_calls	The number of outgoing calls per day.
Missed_calls	The number of missed calls per day.
Inc_talk	The total amount of time spent talking on the phone (call duration) per day (in seconds) including both incoming and outgoing calls.
Out_talk	The amount of time spent talking (in seconds) per day on the incoming calls.
All_call_time	The amount of time spent talking (in seconds) per day on the outgoing calls.
Inc_call_time	The time that calls either received or made, ranging between 0 and 24 (0AM – 12PM).
Out_call_time	The arrival time of incoming calls, 0-24 (0AM – 12PM).
Ent_call_time	The departure time of outgoing calls, 0-24 (0AM – 12PM).

The last feature in Table 1 and 2 is information entropy [2] which is a measure of the uncertainty of a random variable. In our case, this random variable is calling time. The entropy of a variable X is defined by (1) where $x_i \in X$ and $P(x_i) = Pr(X=x_i)$.

$$H(X) = -\sum_i P(x_i) \log_2(P(x_i)). \tag{1}$$

Assessment based on these extracted features is that Sloan students tend to make more phone calls than Media Lab students, whereas Media Lab students like to talk (or spend time) on the phone longer on outgoing calls but talk less on incoming calls than Sloan students. Sloan students spend time on the phone later in the day (about 2:30PM) than Media Lab students (about 1PM). Lastly, the randomness in calling time of Media Lab students is higher than Sloan students.

5 Feature Selection

So far, we have extracted features from raw data (call logs) and we need to select the useful features for classification. This section discusses how to evaluate the usefulness of features for classification. In general, for classification task as we try to assign an unknown sample to different classes which have different characteristics. Our goal is to find a character (e.g. a set of features) of the unknown sample that mostly identifies its belonging class among other classes. This set of features need to have high degree of difference (or low degree of similarity) to other classes to be considered as a “good” set of features. If we adopt the correlation between two random variables as a goodness measure, the above definition becomes that a feature is good if it is highly correlated with the belonging class but not highly correlated with other classes.

There are two main approaches to measure the correlation between two random variables. One is based on classical linear correlation and the other is based on information theory. The first approach is the well known *linear correlation coefficient* (r). For any pair of random variables (X, Y) , r can be computed by (2).

$$r = \frac{\sum_i (x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sqrt{\sum_i (x_i - \bar{x}_i)^2} \sqrt{\sum_i (y_i - \bar{y}_i)^2}}, \tag{2}$$

where \bar{x}_i is the mean of X , and \bar{y}_i is the mean of Y . The value of r is between -1 and 1. A correlation coefficient of 1, -1, and zero implies perfect linear relationship, inversely proportional relationship, and no linear relationship between the two variables respectively. It is symmetrical measure for two variables. There also exists other measures in this category which are basically variations of r , such as *least square regression error* and *maximal information compression index* [3]. There are several benefits of choosing linear correlation coefficient as a goodness measure for feature selection such as it helps remove features with correlation close to one from selection and retain other features with low correlation. However, in the reality it is not safe to always assume “linear” relationship between features. Linear correlation measures may not be able to capture the correlations that are not linear in nature.

Another approach to measure the correlation which is based on information theory can overcome this shortcoming. We adopt the concept of information entropy which is given in (1) which measures the degree of uncertainty between two random variables. Information theory [4] defines conditional entropy of a random variable given another with a joint distribution $P(x_i, y_j)$ as follows.

$$H(X | Y) = -\sum_i \sum_j P(x_i, y_j) \log_2(P(x_i | y_j)). \tag{3}$$

Another important definition is *mutual information* which is a measure of the amount of information that one random variable contains about another random variable which is given by (4).

$$I(X; Y) = -\sum_i \sum_j P(x_i, y_j) \log_2\left(\frac{P(x_i, y_j)}{P(x_i)P(y_j)}\right). \tag{4}$$

Given (1) and (3), it is straightforward to derive (5).

$$I(X;Y) = H(X) - H(X|Y). \quad (5)$$

Mutual information is also referred to as *information gain* [5] which can be interpreted as a measure of the amount by which the entropy of X decreases reflects additional information about X provided by Y .

Theorem. The mutual information is symmetrical for two random variables X and Y which can be proved as follows.

Proof. To show that $I(X;Y) = I(Y;X)$.

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= H(Y) - (H(X,Y) - H(X)) \\ &= H(X) + H(Y) - H(X,Y) \\ &= H(X) - H(X|Y). \end{aligned}$$

For fairness in comparisons, normalization is needed. Therefore the *normalized mutual information* can be derived as

$$I_{Nor}(X;Y) = \frac{H(X) - H(X|Y)}{H(X)}, \quad (6)$$

where denominator $H(X)$ is a scale factor to normalize it to $[0, 1]$.

6 Kernel-Based Naïve Bayesian Classifier

The approach to classification taken here is based on Bayes rule [6] of conditional probability which is given by (7).

$$P(Y|X) = P(Y) \frac{P(X|Y)}{P(X)}, \quad (7)$$

where $P(Y|X)$ is the *a posteriori* probability which is the probability of the state of nature being Y given that feature value X has been measured. The *likelihood* of Y with respect to X is $P(X|Y)$ which indicates that other things being equal, the category Y for which $P(Y|X)$ is large is more “likely” to be the true category. $P(Y)$ is called *a priori* probability. The *evidence* factor, $P(X)$, can be viewed as a scale factor to guarantee that the posterior probabilities sum to one.

Suppose now that we have N input features, $X = \{x_1, x_2, \dots, x_N\}$, which can be considered independent both unconditionally and conditionally given y . This means that the probability of the joint outcome x can be written as a product,

$$P(X) = P(x_1) \cdot P(x_2) \cdots P(x_N) \quad (8)$$

and so can the probability of X within each class y_j ,

$$P(X|y_j) = P(x_1|y_j) \cdot P(x_2|y_j) \cdots P(x_N|y_j). \quad (9)$$

With the help of these it is possible to derive the basis for the *naïve Bayesian classifier* [7] as follows,

$$P(y_j | X) = P(y_j) \frac{P(X | y_j)}{P(X)} = P(y_j) \prod_{i=1}^N \frac{P(x_i | y_j)}{P(x_i)}. \tag{10}$$

The designation *naïve* is due to simplistic assumption that different input attributes are independent.

From (10), the classification is then based on the likelihood function given by (11).

$$L(y_j | X) = \prod_{i=1}^N P(x_i | y_j). \tag{11}$$

Most applications that apply naïve Bayesian classifier derive likelihood function from the actual data or assumed parametric density function (e.g. Gaussian, Poisson). Another approach to derive likelihood function is by using non-parametric density estimation. The most popular method is the kernel estimation which is also known as the Parzen window estimator [8] as follows,

$$f(z) = \frac{1}{Mh} \sum_{k=1}^M K\left(\frac{z - z_k}{h}\right), \tag{12}$$

where $K(u)$ is kernel function, M is the number of training points, and h is the bandwidth or smoothing parameter. The most widely used kernel is Gaussian of zero mean and unit variance ($\mathcal{N}(0,1)$) which is defined by (13).

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}. \tag{13}$$

The choice of the bandwidth h is crucial. Several optimal bandwidth selection techniques have been proposed ([9]). In this study, we use AMISE optimal bandwidth selection using the *Sheather Jones Solve-the-equation plug-in* method which was proposed in [10].

Kernel density estimator provides smoothness to likelihood function with continuous attributes rather than relying on discrete ones. Now the likelihood function in (11) becomes

$$L(y_j | X) = \frac{1}{Mh} \prod_{i=1}^N \left(\sum_{k=1}^M K\left(\frac{y_j - z_k^i}{h}\right) \right), \tag{14}$$

where z_k^i is training point k of feature i .

7 Implementation and Results

To evaluate our proposed system, we continue our implementation from data extraction process in section 4. Recall that we have 11 extracted features from the call logs. Now we need to select useful features based on normalized mutual information as discussed in section 5. Based on (6), normalized mutual information is computed for each feature and plotted in Fig. 2 for comparison. If normalized mutual information of 0.5 is chosen as a threshold, then we have six featured selected with the highest degree of discriminancy.

The six selected features with their corresponding normalized mutual information are listed in ascending order in Table 3. Recall that less normalized mutual information implies higher order of discriminancy or most useful feature for classification.

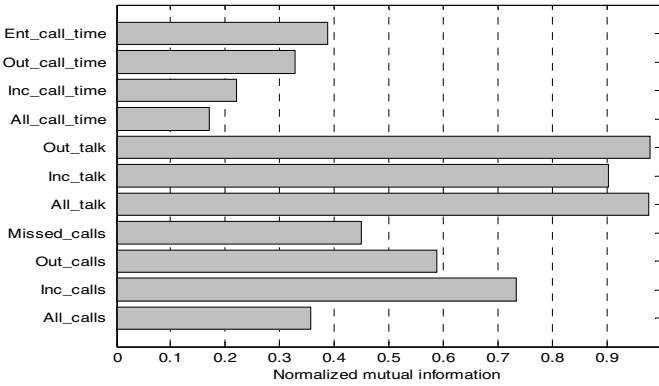


Fig. 2. Result of normalized mutual information

Table 3. Selected features based on normalized mutual information

Features	Normalized Mutual Information
All_call_time	0.169
Inc_call_time	0.220
Out_call_time	0.328
All_calls	0.357
Ent_call_time	0.388
Missed_calls	0.450

The useful features have been selected, before reaching classifier feature normalization is needed. The reason for normalization is to reduce the noisiness of features since non-normalized features have different ranges and are measured in different units. Thus, selected features are normalized to [0, 1].

Features are now ready to be fed to classifier which operates in two modes; training and testing. We use 50% of our feature set as training data and the other 50% as testing data. We implement our proposed method of using kernel-based naïve Bayesian classifier with selected six features based on normalized mutual information. The performance of our proposed method is measured by the accuracy rate which is a ratio of correct classified users to the total testing users. For performance comparison purposes, we also implement naïve Bayesian classifier using all 11 extracted features, naïve Bayesian classifier using six selected features, and kernel-based naïve Bayesian classifier using all 11 extracted features to compare with our method. The result is shown in Table 4, among four approaches, our approach has the best performance with accuracy rate of 81.82%. Naïve Bayesian classifier using all 11 extracted

Table 4. Accuracy comparison of classifier with different methods

Methods	Accuracy Rate (%)
Naïve Bayes with all features	59.09
Naïve Bayes with six selected features	68.18
Kernel-based naïve Bayes with all features	77.27
Kernel-based naïve Bayes with six selected features	81.82
Naïve Bayes with all features	59.09
Naïve Bayes with six selected features	68.18

features, naïve Bayesian classifier using six selected features, and kernel-based naïve Bayesian classifier using all 11 extracted features perform at accuracy rates of 59.09%, 68.18%, and 77.27% respectively.

In addition, to evaluate the effectiveness of the six selected features based on normalized mutual information, we sort all 11 features based on normalized mutual information in ascending order and monitor the changes in accuracy rate as more ascending sorted features taken into account. We monitor both kernel-based naïve Bayesian and classical naïve Bayesian approach which are shown in Fig. 3. Accuracy rate of both methods continue to increase up to when six features are taken into account, then accuracy rate decreases. The accuracy rate continues to decrease after more than six features taken for naïve Bayesian classifier whereas the accuracy decreases from six to seven features and stays constant until all 11 features are taken into account for kernel-based naïve Bayesian classifier.

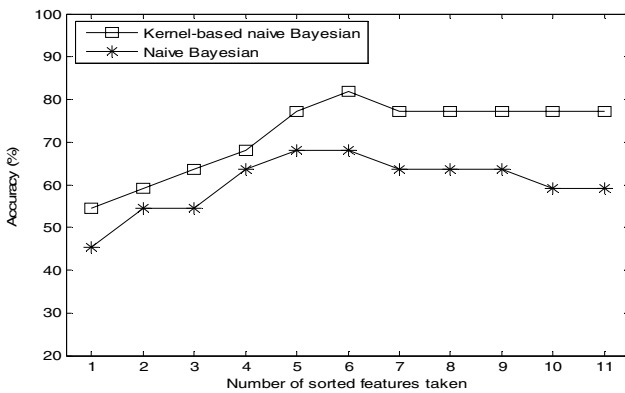


Fig. 3. Change of accuracy according to number of features selected

Figure 3 tells us that the selected six features listed in Table 3 are indeed useful features for classification. Including more features for classifier does not mean better performance. In fact, it may degrade the performance of classifier with its noisiness and low degree of discriminancy.

8 Conclusion

According to the CTIA [11], there are currently 243 million mobile phone subscribers in the US. With a current population of around 300 million and assuming that the CTIA figure implies unique subscribers, about two in every three Americans own a mobile phone. With this widespread use of mobile phones, it becomes valuable source of information for social networks analysis. In this paper, we analyze social networks based on mobile phone's call logs, and propose a model for inferring groups. We describe data pre-processing process which consists of data extraction and feature selection in which we introduce a technique for selecting features using normalized mutual information that measures degree of discriminancy. With its symmetrical and linearity-invariance property, we show that it makes normalized mutual information suitable for our feature selection process. We adopt the classical naïve Bayesian learning and introduce kernel density estimator to estimate the likelihood function which improves accuracy of the classifier with its smoothness. Our model is evaluated with real-life call logs from Reality Mining project group. The performance is measured by the accuracy rate. The results show that our model performs at accuracy rate of 81.82% which is highest among other models (Naïve Bayesian classifier using all extracted features, naïve Bayesian classifier using six selected features, and kernel-based naïve Bayesian classifier using all extracted features). We believe that our model can be also useful for other pattern recognition and classification tasks. As our future directions, we will continue to investigate on the features that can be extracted from call logs which can be useful for classification. We will also explore other statistical learning techniques to improve accuracy of our model.

Acknowledgements. This work is supported by the National Science Foundation under grants CNS-0627754, CNS-0619871 and CNS-0551694.

References

1. Eagle, N., Pentland, A.: Reality Mining: Sensing Complex Social Systems. *Journal of Personal and Ubiquitous Computing* 10(4), 225–268 (2005)
2. Shannon, C.E.: A Mathematical Theory of Communications. *Bell System Technical Journal* 27, 379–423, 623–656 (July and October 1948)
3. Mitra, P., Murphy, C.A., Pal, S.K.: Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 301–312 (2002)
4. Cover, T.M., Thomas, J.A.: *Elements of Information Theory*. Wiley, Chichester (1991)
5. Quinlan, J.: *C4.5: Programs for machine learning*. Morgan Kaufman Publishers, Inc., San Francisco (1993)
6. Duda, R.O., Hart, P.E.: *Pattern Classification and Scene Analysis*. John Wiley, New York (1973)
7. Good, I.J.: *Probability and the Weighing of Evidence*. Charles Griffin, London (1950)
8. Parzen, E.: On estimation of a probability density function and mode. *Annual Mathematical Statistics* 33(3), 1065–1076 (1962)

9. Wand, M.P., Jones, M.C.: Kernel Smoothing. Chapman & Hall, Boca Raton (1994)
10. Sheather, S.J., Jones, M.C.: A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society, Series B*(53), 683–690 (1991)
11. Wireless Quick Facts. CTIA, Ed. (2007),
<http://www.ctia.org/media/index.cfm/AID/10323>

Group Recommendation System for Facebook

Enkh-Amgalan Baatarjav, Santi Phithakkitnukoon, and Ram Dantu

Department of Computer Science and Engineering
University of North Texas, Denton, Texas, 76203, USA
{eb0050, santi, rdantu}@unt.edu

Abstract. Online social networking has become a part of our everyday lives, and one of the popular online social network (SN) sites on the Internet is Facebook, where users communicate with their friends, join to groups, create groups, play games, and make friends around the world. Also, the vast number of groups are created for different causes and beliefs. However, overwhelming number of groups in one category causes difficulties for users to select a right group to join. To solve this problem, we introduce group recommendation system (GRS) using combination of hierarchical clustering technique and decision tree. We believe that Facebook SN groups can be identified based on their members' profiles. Number of experiment results showed that GRS can make 73% accurate recommendation.

Keywords: Social network, recommendation system, decision tree.

1 Introduction

Face-to-face, voice, email, and video communications are traditional medium of interaction between friends, family, and relatives. The traditional medium takes place when two parties had already shared some form of common value: interest, region, family bond, trust, or knowledge of each other. Although, on online social network (SN) two parties initiate communication without the common values between them, they still can freely share their personal information with each other [1]. In the virtual world, joining or creating groups and making friends are a click of a button, which makes online social networking sites, such as Friendster, MySpace, Hi5, and Facebook more and more popular and diverse each day [14]. Therefore, online SN's advantages are user friendliness and flexible in cyberspace where users can communicate with others and create and join groups as their wishes.

Even though flexibility of online SN brings diversity in cyberspace, it can also lead to uncertainty. We took University of North Texas (UNT) SN as a sample for our research. There are 10 main group types, such as business, common interest, entertainment & arts, geography, music, etc. Six of them have over 500 groups, and four of them have range between 61 and 354 groups in each. It is overwhelming to find a group that fits a user's personality. Our study concentrates on identifying inherent groups' characteristics on SN, so that we develop group recommendation system (GRS) to help the user to select the most suitable group to join.

Groups were created to support and discuss causes, beliefs, fun activities, sports, science, and technology. In addition, some of the groups have absolutely no meaningful purpose, but just for fun. Our research shows that the groups are self-organized, such that users with similar characteristics, which distinguishes one group from others. The members' characteristics are their profile features, such as time zone, age, gender, religion, political view, etc, so members of the group have some contributions to their group identity. The group members' characteristics shape characteristic of the group.

Main Contribution: In this paper, we present Group Recommendation System (GRS) to classify social network groups (SNGs). Even though groups consist of members with different characteristics and behaviors, which can be defined by their profile features, as their group size grow, they tend to attract people with similar characteristics [13]. To make accurate group recommendation, we used hierarchical clustering to remove members whose characteristics are not quite relevant with majority in the group. After removing noise in each group, decision tree is built as the engine of our GRS. In this paper, we show how decision tree can be applied not only to classifying SNGs, but also used to find value of features that distinguish one group from another. GRS can be a solution to online SN problem with the overwhelming number of groups are created on SN sites because anyone can create groups. Having too many groups in one particular type can bring concern on how to find a group that has members who share common values with you. We believe if more and more members share common values, the group will grow in size and have better relationship. Thus, GRS can be a solution to many SNG issues.

The rest of the paper is organized as follows. In Section 2, we discuss related work done on social network. In Section 3, we describe the architecture and framework of GRS. Section 4 presents the performance of GRS. The paper is concluded with summary and an outlook on future work.

2 Related Work

There has been an extensive number of research efforts focused around modeling individual and group behaviors and structure, but due to its vastness we restrict here to providing only a sample of related research projects. Many researches on social networking have been done in mathematics, physics, information science, and computer science based on properties, such as small-world, network transitivity or clustering, degree distributions, and density ([6],[7],[8],[10], and[11]).

From research in statistics, Hoff et al. [9] developed class models to find probability of a relationship of between parties, if positions of the parties are known on a network. Backstrom et al. [2] has done very interesting research on finding growth of network and tendency of an individual joining a group depends on structure of a group.

3 Methodology

In this section, we cover data collection process, noise removal using hierarchical clustering, and data analysis to construct decision tree. Figure 1 shows basic architecture of the group recommendation system (GRS). It consists of three components: i) profile feature extraction, ii) classification engine, and iii) final recommendation.

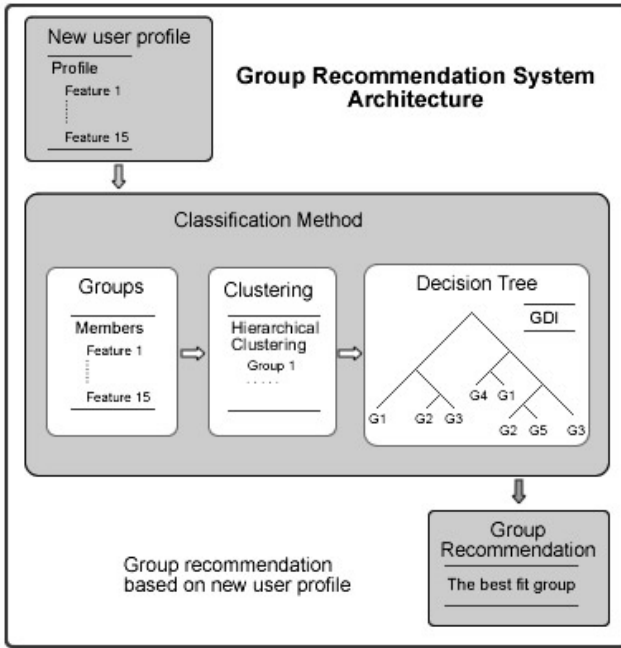


Fig. 1. Basic architecture of GRS, which consists of three major components including profile feature extraction, classification engine, and final recommendation

3.1 Facebook API

The dataset we used in this research was collected using Facebook Platform. Facebook launched its API to public in May 2007 to attract web application developers. The API is available in multiple programming languages: PHP, Java, Perl, Python, Ruby on, C, C++, etc. Since Facebook and Microsoft became partners, Microsoft has launched developer tools in its Visual Studio Express and Propfly. The Facebook Platform is REST-based interface that gives developers access to vast amount of users' profile information.

Using this interface, we had access to student accounts in which privacy setting was configured to allow access to its network (default setting). In our research we used University of North Texas (UNT) social network on Facebook. During this research we were able to access 1580 users' accounts. From the accounts, we collected users' profile information, friend connections, and groups where they belong to. For our analysis, we selected 17 groups from common interest groups on UNT SN. Table 1 shows detailed information of the groups.

3.2 Profile Features

The first step of group recommendation system is to analyze and to identify the features which capture the trend of a user in terms of its interest, social connection, basic information such as age, sex, wall count, notes count and many such features.

Table 1. Information of 17 common interest groups on UNT social network including their subtype categories, number of members, and description

Group	Subtype	Group Size	Description
G1	Friends	12	Friends group for one is going abroad
G2	Politic	169	Campaign for running student body
G3	Languages	10	Spanish learners
G4	Beliefs & causes	46	Campaign for homecoming king and queen
G5	Beauty	12	Wearing same pants everyday
G6	Beliefs & causes	41	Friends group
G7	Food & Drink	57	Lovers of Asian food restaurant
G8	Religion & Spirituality	42	Learning about God
G9	Age	22	Friends group
G10	Activities	40	People who play clarinets
G11	Sexuality	319	Against gay marriage
G12	Beliefs & causes	86	Friends group
G13	Sexuality	36	People who thinks fishnet is fetish
G14	Activities	179	People who dislike early morning classes
G15	Politics	195	Group for democrats
G16	Hobbies & Crafts	33	People who enjoys Half-Life (PC game)
G17	Politics	281	Not a Bush fan

We extracted 15 features to characterize a group member on Facebook: Time Zone - location of the member, Age, Gender, Relationship Status, Political View, Activities, Interest, Music, TV shows, Movies, Books, Affiliations - number of networks a member belongs to, Note counts - number of member's note for any visitors, Wall counts - visitor's note for member's page, Number of Fiends - number of friends in the group.

Based on analysis of 17 groups, we found some interesting results of differences between groups. Figure 2 illustrates gender ratio, age distribution, and political view in 17 groups. It is also useful to draw parallel attention between Table 1 and Fig. 2. G1 is a friend group, and majority of the members are Female, age between 20 and 24, and 33% don't share their political preference. Same 33% are moderate. These properties identify G1. Same way we can interpret all 17 groups. Female members are majority in G1 (friends group), G4 (campaign for homecoming king and queen), G7 (Asian food lovers), G10 (clarinet players), G13 (people who likes fishnet), and G17 (Not Bush fan). At same time, majority of G17 consider themselves as liberal. Fig. 2(b) shows that majority of all groups are members between age 20 and 24. Fig. 2(c) illustrates that majority of G3 (spanish learners), G5 (wearing same pants everyday), G7 (Asian food lovers), G8 (religions group), G10 (clarinet players), G12 (friends group), G16 (PC gamers) did reveal their political preference.

As we can see that using this property, we can construct a decision tree to make better group selection for Facebook users.

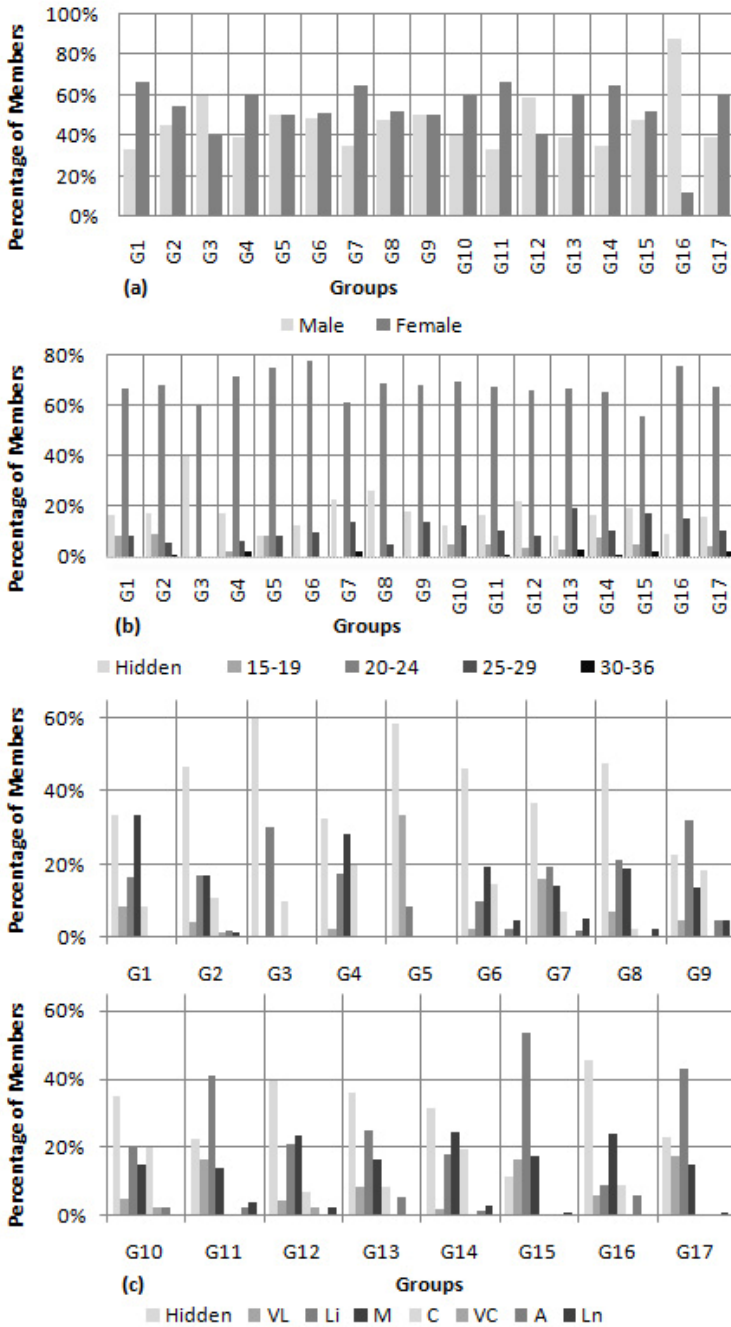


Fig. 2. (a) Gender ratio of each group. (b) Age distribution ranges of 15 to 19, 20 to 24, 25 to 29, and 30 to 36. (c) political preference distribution of the following very liberal (VL), liberal (Li), moderate (M), conservative (C), very conservative (VC), apathetic (A), and libertarian (Ln).

3.3 Similarity Inference

One of the frequently used techniques to find similarity between nodes in multidimensional space is hierarchical clustering analysis. To infer similarity between members, we use Euclidian distance [12].

Clustering takes place in the following steps for each group: i) normalizing data (each feature value = [0, 1]), ii) computing a distance matrix to calculate similarities among all pairs of members based on Eq. (1), iii) using unweighted pair-group method using arithmetic averages (UPGMA) on distance matrix to generate hierarchical cluster tree as given by Eq. (2).

$$d_{rs} = \sqrt{\sum_{i=1}^N (x_r - x_s)^2}, \quad (1)$$

where d is the similarity between nodes r and s , N is number of dimensions or number of profile-features, and x is value at a given dimension.

$$d(r, s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} dist(x_{ri}, x_{sj}), \quad (2)$$

where n_r is number of cluster in r , n_s is number of cluster in s , x_{ri} is the i th object in cluster r , and x_{sj} is the i th object in cluster s . The Eq. (2) finds average distance between all pairs in given two clusters s and r .

Next step is to calculate clustering coefficient to find the cutoff point such that noise can be reduced. In Section 3.4 shows finding clustering coefficient.

3.4 Clustering Coefficient

Each group has a unique characteristic, which differentiates it from others, yet some members within the same group may have different profiles. As these differences grow to some extent, these members emerge as an inevitable “noise” for clustering.

To detect and mitigate this noise thus the group is strongly characterized by core members who establish innermost part of the group, we introduce the *clustering coefficient* (C), which is given by Eq. (3).

$$C = \frac{N_{R_i}}{R_i}, \quad (3)$$

where R_i is the normalized Euclidean distance from the center of member i , given by Eq. (4) hence $R_i = [0, 1]$, and N_k is the normalized number of members within distance k from the center, given by Eq. (5) and hence $N_k = [0, 1]$.

$$R_i = \frac{r_i}{\max_j(r_j)}, \quad (4)$$

where r_i is the distance from the center of member i and $i = \{1, 2, 3, \dots, M\}$.

$$N_k = \frac{n_k}{M}, \quad (5)$$

where n_k is the number of members within distance k from the center, and M is the total number of members in the group.

To reduce the noise in the group, we retain only members whose distances from the center are less than and equal to R^x as shown in Fig. 3, where R^x is the distance at which clustering coefficient reaches the maximum.

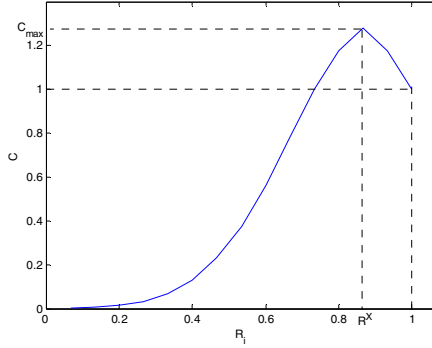


Fig. 3. An example of C -vs- R_i plot for finding R^x . This plot illustrates the cutoff distance R^x which is the corresponding distance from the center at which the clustering coefficient is maximum and gradually decreasing to 1 at $R_i = 1$. As the clustering coefficient starts to decrease, the sparseness of the outer circular members increases more rapidly since the denominator starts to dominate (greater than numerator) according to Eq. (3). Hence the outer circular members or members who have distance from the center greater than R^x to be considered as noise and removed.

3.5 Decision Tree

The nature of group recommendation system (GRS) is classification type problem. Based on a user’s profile features, GRS finds the most suitable groups for a user. One solution to classification type problem is decision tree algorithm, based on binary recursive partitioning. There are number of splitting rules: Gini, Twoing, and Deviance [3]. To find better result we integrated each of splitting rule to GRS. However, test showed no significant improvement in accuracy, which means that final tree does not depend on what splitting rule is used to construct the tree [3]. The main goal of these splitting algorithms is to find the best split of data with maximum homogeneity on each side. Each recursive iteration purifies data until the algorithm reaches to terminal nodes (classes).

Binary tree consists of parent node t_p and child nodes of t_l and t_r . To define maximum homogeneity of child node, we introduce impurity function $i(t)$, so maximum homogeneity of t_l and t_r nodes is equal to the maximum change in impurity function $\Delta i(t)$ (given by Eq. (6)), which shows that splitting rule go through all variable values to find the best split question $x_i \leq x_j^R$, so that maximum $\Delta i(t)$ is found.

$$\Delta i(t) = i(t_p) - P_l i(t_l) - P_r i(t_r), \tag{6}$$

where P_l and P_r are probabilities of left and right nodes, respectively. Thus, maximum impurity is solved on each recursion step and given by Eq. (7).

$$\max_{x_j \leq x_j^R, j=1 \dots M} [i(t) = i(t_p) - P_l i(t_l) - P_r i(t_r)], \tag{7}$$

where x_j is variable j , x_j^R is the best possible variable x_j to split, M is number of variables.

4 Result

In this research we developed group recommendation system (GRS) using hierarchical construct and decision trees. To evaluate the performance of GRS, we used 50% of data for training and other 50% for testing. For testing, we selected labeled members and clustered those using GRS. Accuracy rate is measured by the ratio of correct clustered members to total testing members. Figure 4 compares accuracy of GRS with clustering and without clustering for noise removal. Average accuracy without clustering was 64%. Meanwhile, after removing noise from each group using clustering coefficient method, average accuracy improved to 73%. In other words, average accuracy improved by 9%. In addition, 32% of 1580 members or 343 members were found to be noise and eliminated.

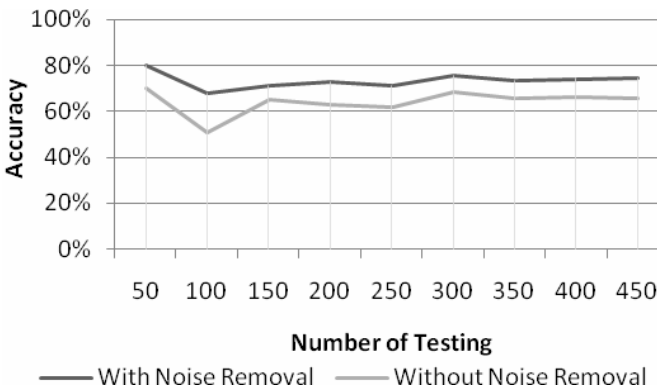


Fig. 4. Accuracy comparison of GRS with and without clustering where the accuracy is improved by 9% with clustering

5 Conclusion and Future Work

It is challenging to find a suitable group to join on SN, especially networks as big as MySpace and Facebook. Until now, online social networking has no sign of slowing down. While Facebook has 42 million users as of October 2007, there are 67 million active users as of February 2008. It has been doubling its size in every six months.

To improve quality of service for Facebook users, we developed GRS to find the most suitable group to join by matching users’ profiles with groups’ identity. The system is built using combination of hierarchical clustering and decision tree. After removing noise, we achieved 9% average accuracy improvement over without removing noise and average accuracy of 73%.

Nature of decision tree is well suited for generating list of most favorable groups for user. In our future work, we will improve the GRS by listing a certain number of most suitable groups according to the users’ profile. Tree figure on Fig. 1 illustrates that once the most suited group is found, other nodes in same sub-tree or neighbor share similarity with the most suited group. This property can be vital to find list of suitable groups.

The main concept behind the GRS can be used in many different applications. One is information distribution system based on profile features of users. As social networking community expands exponentially, it will become a challenge to distribute right information to a right person. We need to have a methodology to shape flooding information to user from his/her friends, groups, and network. If we know identity of the user's groups, we can ensure the user to receive information he/she prefers.

Another research area can be explored is targeted-advertising [4] to individuals on social network site. Many advertising technique are already implemented, such as Amazone based on users' search keywords and Google Adsense based on context around its advertising banner. In addition, Markov random field technique has emerged as useful tool to value network customer [5].

Acknowledgments. This work is supported by the National Science Foundation under grants CNS-0627754, CNS-0619871 and CNS-0551694. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We would like to thank our anonymous reviewers for their insightful and helpful comments and suggestions.

References

1. Adamic, L.A., Buyukkokten, O., Adar, E.: A social network caught in the web. *First Monday*, vol. 8 (2003)
2. Backstrom, L., Huttenlocher, D.P., Kleinberg, J.M., Lan, X.: Group formation in large social networks: Membership, growth, and evolution. In: *KDD*, pp. 44–54 (2006)
3. Breiman, L., Friedman, J., Olshen, R., Stone, C.: *Classification and Regression Trees*. Chapman & Hall, New York (1984)
4. Chickering, D.M., Heckerman, D.: A decision theoretic approach to targeted advertising. In: *UAI*, pp. 82–88 (2000)
5. Domingos, P., Richardson, M.: Mining the network value of customers. In: *KDD*, pp. 57–66 (2001)
6. Flake, G.W., Lawrence, S., Giles, C.L., Coetzee, F.: Self-organization and identification of web communities. *IEEE Computer* 35(3), 66–71 (2002)
7. Flake, G.W., Tarjan, R.E., Tsioutsoulis, K.: Graph clustering and minimum cut trees. *Internet Mathematics* 1(4), 385–408 (2004)
8. Girvan, M., Newman, M.: Community structure in social and biological networks. *PNAS* 99(12), 7821–7826 (2002)
9. Hoff, P.D., Raftery, A.E., Handcock, M.S.: Latent space approaches to social network analysis. *Journal of the American Statistical Association* 97(460), 1090–1098 (2002)
10. Hopcroft, J.E., Khan, O., Kulis, B., Selman, B.: Natural communities in large linked networks. In: *KDD*, pp. 541–546 (2003)
11. Liben-Nowell, D., Novak, J., Kumar, R., Raghavan, P., Tomkins, A.: Geographic routing in social networks. *PNAS* 102(33), 11623–11628 (2005)
12. Romesburg, H.C.: *Cluster analysis for researchers*. Lulu Press, North Carolina (2004)
13. Viegas, F.B., Smith, M.A.: Newsgroup crowds and authorlines: Visualizing the activity of individuals in conversational cyberspaces. In: *HICSS* (2004)
14. Wellman, B., Boase, J., Chen, W.: The networked nature of community. *IT&Society* 1(1), 151–165 (2002)

Towards a Scalable and Collaborative Information Integration Platform and Methodology

Felix Van de Maele¹ and Alicia Díaz²

¹ Collibra nv/sa Ransbeekstraat 230, 1120 Brussels, Belgium

felix@collibra.com

² Lifia, Fac. Informática, UNLP

cc 11, (1900) La Plata, Argentina

alicia.diaz@lifia.unlp.edu.ar

Abstract. The quality and scalability problems in information integration are still not sufficiently solved by current integration approaches. In this paper, we propose an integration approach that adopts a uniform, context-aware, and collaborative methodology enabling efficient reuse to come to a scalable integration platform. We present a two-step methodology that splits up the traditional process of information integration into two separated phases. In the first phase, the Mapping phase, heterogeneous models are matched and mappings are created between the corresponding entities. We further introduce a community and contextual dimension for each mapping. In the second phase, the Commitment phase, a final, application-specific alignment is created in a certain integration format. We argue that this methodology enables a more scalable, efficient and collaborative integration process. We have developed a platform which is based on this methodology and have also done a preliminary evaluation by going through a real use case.

1 Introduction

In our information society, the need and demand for high-quality information is constantly increasing. As these information sources are heterogeneous and distributed by nature, the need to integrate this enormous amount of data is becoming ever more important. Since long, a lot of research has been done to allow us to cope with the overflow of information around us. However, in our ever more connected and fast-moving society, a more dynamic, adaptive and agile approach is needed to support the rapidly changing requirements and use of information for different stakeholders. These trends and shifting needs have led enterprises to adopt a Service Oriented Architecture (SOA) and is reflected in the OMG's Model Driven Architecture¹ (MDA) initiative. Furthermore, the recent trend towards and acceptance of the need for collaboration and what is often referred to as "The Wisdom of Crowds" has made us believe novel approaches should be explored to address the increased and shifted need for information integration.

In our research, we have identified several shortcomings in existing integration approaches. More specifically, the lack of a scalable methodology and platform enabling

¹ <http://www.ibm.com/developerworks/rational/library/3100.html>

the integration of a very large amount of data sources by a very large number of stakeholders made us question the efficiency of these approaches for real-world scenarios. The moderate support for collaboration and evolution has furthermore motivated us to propose a novel approach for model and data integration: We propose in this paper an *integration approach that adopts a uniform, context-aware, and collaborative methodology enabling efficient reuse to come to a scalable integration platform*.

This paper is organized as follows. Section 2 provides a brief background on semantic integration and discusses some of the existing works. In section 3 our approach and methodology are presented. Section 4 is dedicated to the preliminary evaluation by going through a real use case: the integration of two ontologies and section 5 briefly introduces the basic ideas of the developed platform. Finally, there are conclusions.

2 Background

Integrating two heterogeneous models (e.g. OWL ontologies, Dogma Lexons [2], UML diagrams, relation database schemas, etc.) corresponds to finding a set of mappings between the entities of the first model and the entities of the second model that enable the first model to be translated into the other, and/or the other way around. In the current literature, the integration process of finding a final set of mappings between two models is considered one process within a well-defined time span. Some authors use the notion of Match Operator to describe the mapping process [6], while other authors talk about an alignment process [1]. Both approaches adopt a one-step process (or several chained one-step processes) without any persistent state between the start and the end of the process. In this paper, we argue that

1. In order to achieve a scalable and collaborative integration process, mapping and alignment reuse is very important;
2. Effective and efficient reuse requires the integration process to be split up into two separate phases;
3. A persistent and application-neutral state is necessary between these two phases;
4. Introducing a context and community dimension will significantly increase the efficiency of the process and quality of the resulting alignments.

There are different matching systems [7] which describe an information integration process. These systems actually compute the mapping using a combination of matching algorithms. Most of the existing integration frameworks do not explicitly support mapping reuse. Currently, a mapping is the interpretation of a relation by a single individual for a specific integration problem. There are no processes to represent the interpretation and agreement of a group of stakeholders. Furthermore, as these mappings are not shared, every individual must redo the entire mapping process for integration scenarios that may be very similar to already solved problems.

Existing approaches also make no distinction between the conceptual meaning of a mapping and its application specific representation. Following the definition of semantic integration, mappings must allow one model to be translated into the other,

hence the mappings between the entities of these models need to be described in a specific integration language or integration model that can be interpreted and executed by a specific application for a specific purpose. This limits the applicability of the alignment to the applications that can execute it. Another drawback of directly working with language- or application-specific mappings is that only an actor that is familiar with the language or application can create these mappings. The specific domain knowledge needed to grasp the correspondences between heterogeneous models is often only known by domain experts and users with a managerial role in the organisation, which may not be familiar with the specific integration language.

These issues are critical for an integration approach focusing on mapping reuse: To be able to establish a shared, reusable knowledge contained within the mappings, the involved stakeholders must be able to communicate in a manner that is understood by all participants. Therefore, it is important that these shared mappings are expressed on a conceptual level that is also understood by non-technical knowledge and domain experts.

3 Methodology

In this paper we propose a possible solution for the limitations of existing integration inspired by the STARLab Dogma approach [2] to ontology engineering and OMG's MDA paradigm. We argue for an integration approach that splits up the integration process into two separated phases: the *Mapping Phase* and the *Commitment Phase*. In short, the Mapping Phase takes two heterogeneous models as input and returns a set of mappings between the entities of both models. The difference lies in the fact that these mappings are application and paradigm-neutral and have an additional context and community dimension. We call these mappings *plausible mappings* as they might be valid in a certain context and community, but not necessarily in another. In the second phase, the Commitment Phase, a meaningful subset of plausible mappings is chosen depending on the context, the involved community, and the application domain of the particular integration scenario. This set of application-specific mappings is the final alignment in a certain integration language. We call this process: *committing a set of plausible mappings to an application-specific alignment*.

3.1 Mapping Phase

The goal of the mapping phase is to identify plausible mappings between entities from the heterogeneous models. The result of this process is a set of uniform, plausible mappings that are added to a large mapping base. This process is split up into two methodological steps:

1. *Preintegration*: the syntactical heterogeneities of the models are overcome by parsing the entities of both models into our uniform mapping data model: the Entity/Model. One or more entities from a first model can be mapped to one or more entities from a second model. No syntactic or semantic constraints are put upon these entities.

2. *Matching*: the different entities are matched and mappings are created between similar entities. This matching process can be completely automated by using a set of

matching algorithms [1], but more realistically, matching algorithms will propose a number of mappings after which a human or community should validate these. The resulting mappings are stored in a large mapping repository.

The figure 1.a shows the different methodological steps of the mapping phase. Each mapping process is done within a certain context and by a certain community. Therefore, each mapping stored in the mapping store has a context and community dimension.

3.1.1 Community-Driven and Context-Aware Mappings

By a community we mean here a group of individuals that have a common interest, share a common goal, and follow a number of processes to reach that goal. Recent research has identified a high importance of direct involvement of humans and communities in ontology management: An individual was shown to be an indispensable part of a semantic network [5], and participation of a community has since long been shown as a way to achieve a more up-to-date and complete domain knowledge representation ([2], [8]). We argue that in order to achieve an efficient mapping reuse and hence efficient integration process, mappings must have a community dimension.

In order for stakeholders to reuse mappings they must know from which user and community this mapping was originated. This is important as each community has a certain goal and process which may have influenced the elicitation or creation of the mapping. A mapping might be plausible in a certain community but not in another. We call these *subjective mappings* following earlier work from [8].

Introducing mapping reuse by different users implies solving typical groupware issues like quality, confidence, and trust. It is therefore important that each community can express its own confidence and trust in mappings as these measures might vary between communities.

The community dimension of a mapping can be also used as a grouping mechanism allowing more efficient lookup of mappings in the commitment phase.

On the other hand, the *context* of the mapping models the environmental settings in which the mapping has been deduced. A mapping might be plausible in a certain context, while the same mapping might not be valid in another. We identified three reasons why a contextual dimension is important for a plausible mapping:

Context to disambiguate mappings: the community dimension helps to specifically disambiguate on the creator of the mapping. This is particularly important in our approach to facilitate mapping reuse. A contextual dimension can greatly help the user to find the relevant mappings for his integration problem.

Context to disambiguate relations: In ontology management, the context is used to disambiguate similar terms that refer to different concepts [3]. Context can be similarly used in ontology integration where a relation term in one context refers to a certain relation definition while the same relation term in a different context may refer to another relation definition.

Context to support collaboration: A collaboration environment such as the mapping repository introduces typical groupware problems: What to do if two actors edit the same mapping at the same time and store it back into the repository. One possible solution is to view the problem from a database perspective and use locking techniques.

However, this may hinder the scalability of the platform. Another solution is to build a versioning or merge operator. In this solution, the contextual dimension may be used to automatically create the mapping in a new context when storing a new version or when two mappings cannot be automatically merged. In this way, evolution of mappings can also be supported.

3.2 Commitment Phase

The goal of the commitment phase is to create an *application-specific alignment* that best approximates its intended use. The alignment is modelled in an alignment model described in a certain language to be used by a specific application.

During the commitment process an (group of) actor(s) selects a meaningful set of mappings. This selection and reuse process of committing a mapping to an alignment corresponds to augmenting the representation-neutral mapping to a commitment rule that is valid in this particular alignment described in a particular language. The validity of the commitment-rule is dependent on the intended use of the alignment, which will be domain-, community-, organisation-, or application-specific. Hence, a commitment or alignment can be seen as an interpretation of a set of mappings.

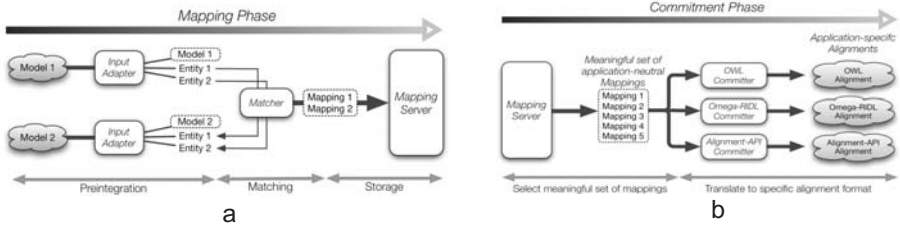


Fig. 1. a) The steps of the Mapping Phase. b) The steps of the Commitment phase.

The resulting alignment corresponds to an instance of an alignment model which is described in a specific language. For example, when integrating two OWL-ontologies, the alignment may be an instance of an OWL Alignment Ontology which provides a language-specific interpretation of the mappings from the alignment. The resulting alignment is language specific and should be created with a certain application scenario in mind. It is also the interpretation of the conceptual mapping from which it originated. In figure 1.b, the methodological steps of the Commitment phase and the necessary tool support are shown.

3.3 Mapping Semantics

This section introduces the specific semantics underlying our methodology and platform. We will provide definitions of the different concepts we have introduced in the previous sections. We extend Euzenat’s model [4] to introduce the notions of entities and models to support the uniform mappings, *context* and *community*.

We start by defining our data model: the *Entity model*. With data model, we understand here the models and the content of the models that are being integrated by the

framework. To enable the integration of heterogeneous data models, we parse any input data model into a model entity combination. The goal is not to describe or specify the full model; we only store a description of it, the type of the model, a reference to the actual model, and the entities that are part of the model.

An *Entity Model* is represented as a 4-tuple: $\langle \text{ref}, \text{desc}, \mathcal{T}, \mathcal{E} \rangle$, where: *ref* is the reference identifier to the actual model, this should be a URI; *desc* is the description of the model; \mathcal{T} is the type of the model (eg. RDFS, OWL, Dogma-Lexon, Relational Database Schema, ...); $\mathcal{E} = \{e_1 \dots e_n\}$ is the set of entities e_1 to e_n that the model contains.

An entity can be anything that is part of the model. It can be a class, a relation, a property, an instance, etc. We have defined an Entity in a similar way as the Entity Model. The entity itself is not stored. We have defined it as such:

An entity e is defined as a 3-tuple: $\langle \text{ref}, \text{term}, \mathcal{Y} \rangle$, where *ref* is the reference identifier of the entity; *term* is the representational term of the entity; \mathcal{Y} is the model of which the entity is part.

Before giving a formal definition of the mapping elements, we first give an informal, pragmatic definition of the two concepts: *community* and *context* in order to agree on a certain terminology:

A *community* \mathcal{P} is modelled as a 3-tuple: $\langle \mathcal{A}, \mathcal{G}, \mathcal{Y} \rangle$ where \mathcal{A} is the set of actors; \mathcal{G} is the goal of the community; and \mathcal{Y} is the process the community wants to follow.

We do not define a precise model for the context. At this time, the context is still a black box. The actors are free to model it as they wish. The goal of the context however, is to model the timeframe, algorithms used, domain, etc. in which the creation of the mapping element took place.

A *context* is represented as the tuple: $\langle \text{ref}, \text{term}, \text{description} \rangle$ where *ref* is the unique reference to this particular context; *term* is the terminological name of the context; and *description* is the informal description of the context.

Now, we can move on to give the formal definition of a *mapping element*. The purpose is to allow mappings between very different data sources, which are an important requirement to be able to use this mapping framework in all of the foreseen scenarios.

A *mapping element* \mathcal{M} between 2 ordered lists of entities $\mathcal{E}_1 = \{e_1, \dots, e_x\} \in \text{Ontology } O_1$ and $\mathcal{E}_2 = \{e_{x+1}, \dots, e_y\} \in \text{Ontology } O_2$ as a 6-tuple $\langle \mathcal{P}, \mathcal{Y}, \mathcal{E}_1, \mathcal{E}_2, \mathcal{Rterm}_\chi, n \rangle$ where: \mathcal{P} stands for the community that was responsible for creating the mapping; \mathcal{Y} is the context in which the mapping was created; \mathcal{E}_1 is the ordered list of entities belonging to *Ontology* O_1 ; \mathcal{E}_2 is the ordered list of entities belonging to *Ontology* O_2 ; \mathcal{Rterm}_χ is the relation term. It is the terminological name of the relation and should, in combination with the context, refer to a unique relation definition. When the relation is fuzzy, χ is the strength of the relation; n is the degree of confidence, a measure of trust in the fact that the mapping is appropriate and it will be given by the community.

The mapping element can also be written in a less verbose manner. A mapping between property $\{\text{foaf} : \text{name}\}$ from the FOAF Ontology² and properties $\{\text{person} : \text{first}\mathcal{N}\{\text{ame}\}, \text{person} : \text{middle}\mathcal{N}\{\text{ame}\}, \text{person} : \text{last}\mathcal{N}\{\text{ame}\}\}$ from ebiquity's Person Ontology³, with a fuzzy union relation with strength 0.75 and confidence 0.95.

$\mathcal{P}, \mathcal{Y} : \{\text{person} : \text{first}\mathcal{N}\{\text{ame}\}, \text{person} : \text{middle}\mathcal{N}\{\text{ame}\}, \text{person} : \text{last}\mathcal{N}\{\text{ame}\}\} \text{union}_{.75,.95} \{\text{foaf} : \text{name}\}$

² <http://www.foaf-project.org/>

³ <http://ebiquity.umbc.edu/ontology/person.owl>

A mapping element is a lexical representation of a conceptual mapping between two lists of entities. As the definition of a mapping element shows, the entities are only references to the actual entities from the model. The relation name is only a reference to a conceptual relation definition. A relational name \mathcal{R}_{term} from a mapping combined with a context \mathcal{Y} refers to exactly one unique relation definition \mathcal{R} .

A *relation definition* \mathcal{R} is represented as the tuple: $\langle name, description, definition, properties \rangle$ where: • *name* is the name of the relation definition; *description* is the textual description of the relation; *definition* is the logical description of the relation, in first order or description logic; *properties* are the properties of the relation; including one or more of the following: unidirectional, bidirectional, transitive, reflexive, irreflexive, symmetrical, asymmetrical and antisymmetrical.

4 The Platform in Action

As a preliminary evaluation of this approach, we will go through the entire process of integrating two similar, but heterogeneous RDF-based ontologies. We have chosen two citation ontologies: UMBC Publication Ontology⁴ and the Bibtex citation Ontology⁵. We will integrate these two ontologies by creating an alignment from the Bibtex Ontology to the Publication Ontology. In this case, we have opted for the Alignment API [3] as the format to store the final alignment in. The Alignment API is an API and implementation for expressing and sharing ontology alignments.

In the mapping phase, to integrate these two heterogeneous ontologies, the models must first be translated into our own data model. We have written an input adapter that translates RDF documents into our Model/Entity model. We have created an RDF Import wizard which can be used to import RDF Ontologies. The user provides the wizard with the file or URL of the ontology and the wizard automatically translates it to our own Model/Entity model. When the wizard is finished, the new model is stored on the Mapping Server and can be seen in the Mapping Navigator (see figure 2.a)

For every mapping drawn in the Editor tab, a detailed mapping is shown in the Mapping tab, which shows all the mappings with the given Community and Context. When a user clicks on a mapping, he can specify its details. Most importantly, he can specify to what Relation Definition the Relation Name (string) must point to. The process of selecting a mapping and pointing the “equal” Relation Name to the Equal Relation Definition is depicted in figure 2.b.

When the user has created the mappings and saves the Editor window like in traditional applications, the mappings are stored on the Mapping Server.

Then, an actor can use our Mapping Client to make the final alignment -- commitment phase. The client can support many different committers depending on the alignment format needed. In this scenario, we will use the Alignment API Committer. This committer will enable the user to select the mappings from the Mapping Server and drag them to the alignment. In the same time, the *committer* will interpret the Relation Definition from these mappings and translate the mappings into the Alignment API format. This process is depicted in figure 2.c.

⁴ <http://ebiquity.umbc.edu/ontology/publication.owl>

⁵ <http://zeitkunst.org/bibtex/0.1>

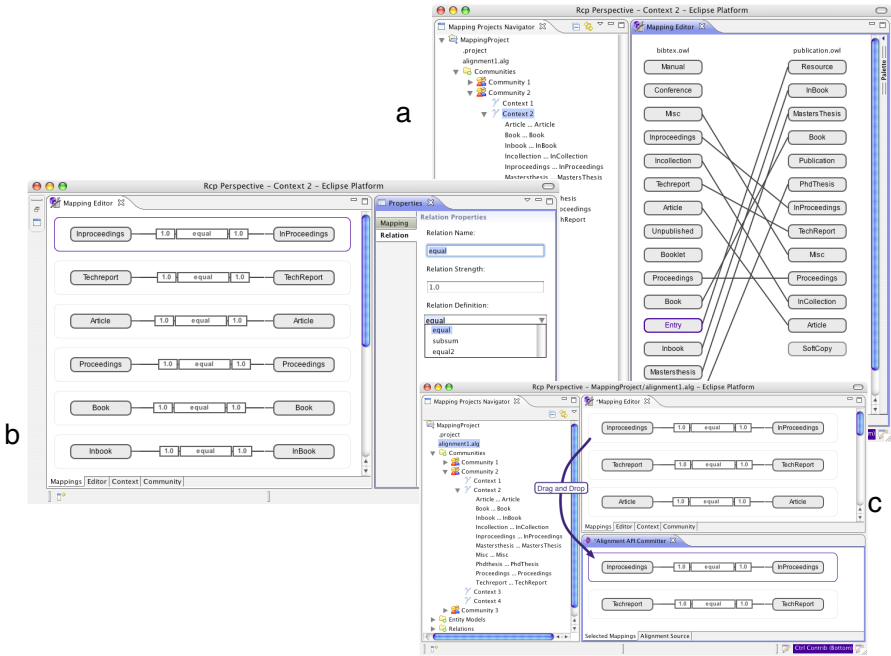


Fig. 2. a) on the left the Mapping navigator shows the result of opening an Editor View and drag and dropping the two models into the Editor tab and drawing mappings between the entities of both models. b) The Mapping tab: detailed mappings are shown and properties can be edited. c) The Selected Mappings tab from the Alignment API Committer plug-in.

The Alignment API Committer has two tabs: the *meaningful mappings* tab (selecting the relevant and meaningful mappings is done by dragging them from the Mapping Editor to this tab window) and the *alignment source* tab (it shows the final alignment, which results from interpreting the selected mappings and converting them to the specific alignment format).

5 Implementation of the Integration Platform

In this section, we will present the architecture of our platform that implements this methodology. As we have discussed, the two cornerstones of our proposed integration approach are mapping reuse and application-specific mappings on our platform, we have split-up the platform into two layers in two orthogonal dimensions. In the first dimension we split-up the platform into a Remote Layer and a Client Layer. In the second dimensions we map our two step methodology, the Mapping phase and the Commitment phase upon the Client Layer. Furthermore, a mediator is needed that mediates between the server and the clients and provides the actors with the necessary services to manage and leverage the existing mappings.

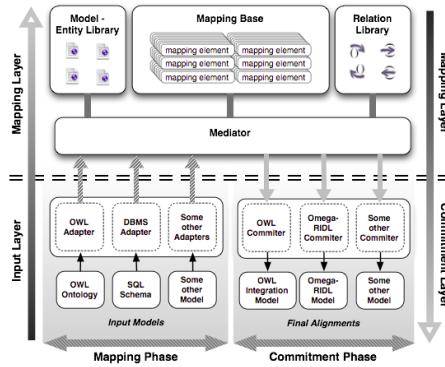


Fig. 3. The two orthogonal dimensions of separation in our platform architecture

To allow public mapping reuse and collaboration, the mappings must be stored on a server that can be accessed by the stakeholders by means of one or more clients. Therefore, we have split up our platform into the two horizontal layers. In the *Remote Layer*, the mappings created in the mapping phase are stored on a Mapping Server. A Mediator is needed to enable the communication from different types of clients with this mapping server. Furthermore, the mediator provides an extra layer of services on top of the simple storage of mappings offered by the mapping server. The *Client Layer* contains all the applications that the involved actors will use to interact with the mappings stored on the remote layer. We can identify two kinds of interactions with the remote layer corresponding to the second dimension and similarly to our two-step integration methodology.

6 Conclusions

We have presented a two-step methodology that splits up the traditional process of information integration into two completely separated phases. In the first phase, the *Mapping phase*, heterogeneous models are matched and mappings are created between the corresponding entities. The mappings are stored on a remote server in an application-neutral manner. We further introduce a community and contextual dimension for each mapping. In the second phase, the *Commitment phase*, a final application-specific alignment is created in a certain integration format or model. We argue that this methodology tackles some of the problems of existing works in the semantic integration domain: *scalability, efficiency and evolution*.

Semantic integration is an expensive process which is currently not scalable: for every integration scenario the entire integration process must be redone. Having the integration process split in two phases, the mapping phase must only be done once. After the plausible mappings are established and store in an application-neutral manner, they can be reused by selecting the relevant mappings and committing them to the application-specific integration language. In that way, the full integration process becomes much shorter, hence faster and more scalable.

The separation of the standard integration process into our two phases can be leveraged to increase the performance and efficiency of the integration process in many ways. To increase the efficiency of the integration process allows the development of platform independent models which can afterwards be transformed into platform specific models. Furthermore, because the created mappings are context-aware and community-driven, they represent the domain and connection with other domains more comprehensibly than the alignments created by the traditional integration approaches.

Our approach facilitates the evolution. When the application in which the alignment must be used changes, only the second phase of our integration methodology must be redone: The mappings stored in the repository can be reused to commit to the new alignment format and the actor must only create new mappings for the new or changed entities.

References

1. Bouquet, P., Ehrig, M., Euzenat, J., Franconi, E., Hitzler, P., et al.: D2.2.1 specification of a common framework for characterizing alignment. KnowledgeWeb Deliverable (February 2005)
2. de Moor, A., De Leenheer, P., Meersman, R.: DOGMA-MESS: A meaning evolution support system for interorganizational ontology engineering. In: Proc. of the 14th Int. Conf. on Conceptual Structures (ICCS 2006), Aalborg, Denmark. LNCS, Springer, Heidelberg (2006)
3. De Leenheer, P., de Moor, A.: Context-driven Disambiguation in Ontology Elicitation. In: Context and Ontologies: Theory, Practice and Applications, AAAI Technical Report WS-05-01, pp. 17–24. AAAI Press, Menlo Park (2005)
4. Euzenat, J.: An API for ontology alignment. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 698–712. Springer, Heidelberg (2004)
5. Mika, P.: Ontologies are us: A unified model of social networks and semantics. *Web Semantics: Science, Services and Agents on the World Wide Web* 5(1), 5–15 (2007)
6. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. *The VLDB Journal* 10(4), 334–350 (2001)
7. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. *Journal on Data Semantics IV* (2005)
8. Surowiecki, J.: *The Wisdom of Crowds*. Anchor (August 2005)

Methodological Approach to Determine Appropriately Annotated Resources in Narrow Folksonomies

Céline Van Damme¹, Stijn Christiaens², and Damien Trog²

¹ MOSI

Vrije Universiteit Brussel

`celine.van.damme@vub.ac.be`

² Semantics Technology and Applications Research Laboratory

Vrije Universiteit Brussel

`{stijn.christiaens,damien.trog}@vub.ac.be`

Abstract. Folksonomies are community managed vocabularies, and do not limit end users to employ a strict terminology in their annotating activities. Users are free to create and use whatever tag they like. Folksonomies have also been criticized to produce low quality meta data due to reduced quality control. In the case of narrow folksonomies where resources are evaluated by only one person there is no certainty that the resources are *appropriately annotated*. In this paper, we suggest a three-phase iterative approach to determine the properties, expressed in terms of tag ambiguity, of resources *appropriately annotated* in a narrow folksonomy to improve information retrieval. We also show brief results of the first steps of that approach in a case study involving a narrow folksonomy.

Keywords: narrow folksonomies, quality, tags, information retrieval.

1 Introduction

Folksonomies do not limit end users to use a predefined terminology to annotate a resource. There is no built-in control mechanism which forbids users to use a certain keyword: they can use whatever tag or keyword that enters their mind. Users are sometimes inclined to tag for personal usage, i.e. idiosyncratic tagging, instead of tagging for general community retrieval purposes [5111]. Idiosyncratic tagging (e.g. *toread*) can in some cases be useful to the whole community (e.g. many people can say that they have *toread* a certain book when annotating a book at Librarything¹). However, this is not always the case, for instance, a picture annotated with *ourhouse* will not always imply that it is a picture of everyones house.

Folksonomies have also been criticized to produce low quality meta data (ambiguity, no synonyms, general and specialized terms) due to reduced quality control [569]. Low quality tags may impede the information retrieval process: for instance if tags are expressed in very specific terms the chances that someone else who is not an expert in the field will retrieve the resource will probably decrease.

¹ <http://www.librarything.com>

Up till now most web sites that rely on tagging systems provide a community feedback mechanism that gives suggestions to the user. In the case of broad folksonomies, where several people tag the same resource, a community feedback mechanism can provide tag suggestions for the resource in question. For instance when a bookmark is already stored in some other users' bookmark collections, Del.icio.us² recommends popular tags, i.e. tags that are frequently added to the bookmark. The same community feedback does not hold for narrow folksonomies, where a resource is tagged only once (e.g. pictures on Flickr³, blog posts of weblogs or employees' created documents stored on a corporate intranet). In latter cases, the systems can only provide feedback on tags in general. For instance, when a user adds a tag to a specific resource the system can show a list of tags often used in combination with that particular tag. Since it is not compulsory for a user to follow the suggestions of the feedback system, the feedback can not be regarded as a *real* tag quality control mechanism.

In the case of narrow folksonomies where resources are evaluated by only one person there is no certainty that the resources are *appropriately annotated*. We define a resource as *appropriately annotated* when the tags are (1) very well describing the content of the annotated resources, (2) expressed in such a way that they can be understood by a large number of members of the community and (3) the resources are easily retrieved by the community members' through tag search.

We assume that tags added to the resources of narrow folksonomies are not always complete and therefore not *appropriately annotated*. In this paper, we want to find the properties of the resources that are appropriately annotated in order to improve the information retrieval process. To quantify the properties we need a measure and suggest using tag ambiguity, in the sense of having more than one meaning (e.g. apartheid could be considered as unambiguous whereas apple has many meanings), because we assume that resources that are appropriately annotated have mainly unambiguous tags.

In Sect. 2 of this paper, we outline a three-phase iterative approach based on tag ambiguity and community input to find the properties of the tags of the resources that are appropriately annotated in a narrow folksonomy. A small part of the approach are briefly tested in a case study involving a narrow folksonomy (i.e. weblog of a research lab) and discussed in Sect. 3. We provide a discussion and a conclusion in Sect. 4.

1.1 Related Work

Research from Golder and Huberman [5] showed that users are more inclined to tag for a personal usage than for the whole community. Sen et al. [12] analysed tags of the MovieLens⁴ recommendation website by categorizing the tags into three possible categories (factual, subjective and personal) which were derived from the ones originally proposed in [5]. Results showed that 63% of the tags are factual (e.g. facts or concepts), 29% personal (e.g. self-reference such as *myhouse*) and 3% subjective (e.g. opinions) and 3% of the tags were undefined.

² <http://del.icio.us>

³ <http://www.Flickr.com>

⁴ <http://movielens.org/>

The authors in [7] propose to extend tags with a kind of rating mechanism that allows users to add an opinion to the tags they have chosen. The user is suggested to tag a resource with neutral tags and/or tags extended with a positive (people like) or negative context (people do not like) to the tag. The positive or negative context is expressed by assigning a positive (+) or negative (-) sign to the tag. For instance a picture of a Ferrari could be tagged as *car*, *expensive(-)*, *fast (+)*. When counting the tags as well as all the positive and negative signs, a global overview on the tagged resource can be obtained.

Asking users to verify whether a tag is merely descriptive or should be broadened with a positive or negative context implies a higher intellectual input from the user and might be too time-consuming for him. We believe it is better to assign such tasks to community members that are very motivated to do so such as community moderators.

Of course, when applying this mechanism presented in [7] to a narrow folksonomy, it still does not provide a certainty regarding the quality of the tags.

In [6] the authors suggest extending the websites that rely on tagging systems with tools, for instance tools that ask questions to the users regarding the resource they tag. Another option the authors propose is giving a sort of tag education to the users to improve the quality of the tags. They argue that the quality of tags can be improved if the users are trained to tag in a consistent way, e.g. only use singular nouns and stemmed verbs. This naturally impedes one of the most important characteristics of tagging: that users are free to use whatever keyword or tag they like.

In [2] we proposed metrics to automatically find the intersubjective tags, tags that are shared by many members of the community, in the case of a broad folksonomy. Preliminary results show that the *high frequency tags* metric, i.e. tags that are most often chosen by the user for a particular resource, can be considered as a good metric to find the intersubjective tag set. These metrics are not so easy to apply to the situation of a narrow folksonomy since resources are tagged only once and we want to find the properties of tags of resources that are appropriately annotated. Other metrics are therefore required.

We suggest a three-phase iterative approach to find the properties, expressed in terms of tag ambiguity, of resources *appropriately annotated* in a narrow folksonomy. The approach includes the calculation of tag curvature as well as feedback by one or more community moderators in order to calculate the degree of tag ambiguity of appropriately annotated resources. The feasibility of the community moderator role can also be seen in similar situations. For instance, Wikipedia⁵ has a group of administrators that regularly review newly produced content. Also, our community moderator role corresponds with the core domain expert role in the DOGMA-MESS approach for community driven ontology evolution [8].

2 Three-Phase Iterative Approach

In this section we present our approach, which consists of three phases: the community bootstrap phase, the community run time phase and the community information retrieval phase.

⁵ <http://www.wikipedia.org>

2.1 Community Bootstrap Phase

The bootstrap phase starts when there are at least 100 resources recently⁶ tagged with meta data. Based on the available data we try to deduce the properties, expressed in terms of tag ambiguity, of resources *appropriately annotated* in a narrow folksonomy.

As we discuss in the next paragraphs, the phase consists of four steps: (1) an evaluation of the training data set, (2) calculation of the percentage of ambiguous tags per resource, (3) calculation of the limiting value of tag (un)ambiguity and (4) testing.

Step 1: Evaluation of a training dataset. First, we prepare a training data set, which contains a selection of randomly annotated resources and its tags. We then propose to appoint one or more community moderators. Their task is to judge the quality of the produced meta data of the training dataset. The community moderators evaluate whether a resource is tagged appropriately as defined in Sect. 1. All the information is stored for the second step. The next step starts when a certain number of resources, we suggest 50, that are marked as appropriately tagged are reached.

Step 2: Calculation of the percentage of ambiguous tags per resource. Based on the evaluation of the training data set, we try to determine the properties of appropriately tagged resources by calculating the percentage of ambiguous tags. For each resource, the moderator has to specify whether a tag is ambiguous, ambiguous in the sense of having more than one meaning, and this for every tag annotated to the resource. We count the number of ambiguous tags and divide the sum by the total number of tags. By averaging these quotients we obtain the percentage of ambiguous tags per appropriately tagged resources. For instance, an ambiguity percentage of 65% implies that 65% of the total tags annotated to a resource are ambiguous whereas 35% are unambiguous.

Step 3: Calculation of the limiting value of tag (un)ambiguity. Of course, if we want to automate previous step in the long run, we need a measure that allows us to calculate the ambiguity of a certain tag. In [3], the author showed that curvature or the clustering coefficient of a node in a graph, where the edges represent the broader/narrower relations between terms obtained from Wordnet⁷, can be used as a measure for ambiguity. The curvature is calculated by dividing the total number of edges between the node's neighbours by the possible number of links among the nodes neighbours. The node X on the left graph in Fig. 1 has no curvature because there are no edges between ns neighbours. Whereas the node X in the right graph has a clustering coefficient of 0.33. Maximum 6 links are possible among Xs neighbours however only 2 of them actually exist. A low curvature implies that the node's neighbours have a small number of inter linkages and are consequently less ambiguous. The author in [3] proved that the curvature is a better measure than the frequency of words in a document as often suggested in literature.

We believe that the curvature as described in [3] can easily be applied to tags since tags can be visualised in a graph based on the conditional probability of tag couples after tag cleansing and calculation of tag co-occurrence. Co-occurrence is an often-suggested

⁶ All the content must have been created in a time period of one year.

⁷ <http://wordnet.princeton.edu/>

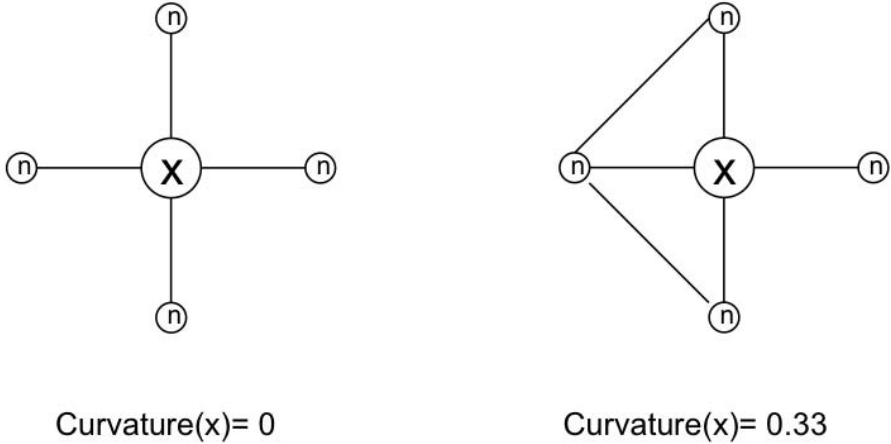


Fig. 1. Example of curvature

analysis technique in literature on folksonomies [13,10] to find relationships between tags. For each tagged resource all the tag pairs are determined. The tie strength between a tag pair is increased each time two tags are used together in other annotated resources. We suggest taking all the tags of the data set into account to calculate the co-occurrence. Since some tags will probably have a very low and other a high co-occurrence we have to set a limiting value when constructing the graph (e.g. the co-occurrence must have a tie strength which exceeds 5).

After calculating the co-occurrence of tags, the broader/narrow relations between the couple of tags has to be calculated by applying the definition of the conditional frequency. The conditional probability is calculated by dividing the co-occurrence of the tag pair by the frequency of the individual tags. Results vary between 0 and 1 [10]. The higher the result, the more the term is used in combination with the other term and consequently the more dependent it is of the other term. A broader/narrow relation is found when a limiting value is exceeded. In [10] the limiting value is set to 0.8.

Then the tags can be plotted into a graph where nodes represent tags and the edges visualize the broader/narrow relation among tags. The curvature has to be calculated for every tag that has been evaluated by a moderator. In case a certain tag is not included in the graph, for instance when the tag has very a low co-occurrence tie strength or the broader/narrow limiting value is much lower.

We calculate the curvature for each tag which was evaluated in the previous step by the moderator as ambiguous (A) or unambiguous (UA). Then, the information (A/UA and curvature) has to be merged into one table to determine the ambiguous/unambiguous limiting values. We take the maximum curvature for the unambiguous limiting value because the closer to 0 the less ambiguous a tag will be. For instance in Fig. 2 we may conclude that the limiting value for tag unambiguity is 0.3. This implies that a tag with curvature value between 0 en 0.3 is unambiguous. To find the ambiguous limiting value we take the minimum value corresponding to ambiguous because the closer to 1 the more ambiguous a tag will be. In Fig. 2 the limiting value for tag unambiguity is 0.55. This

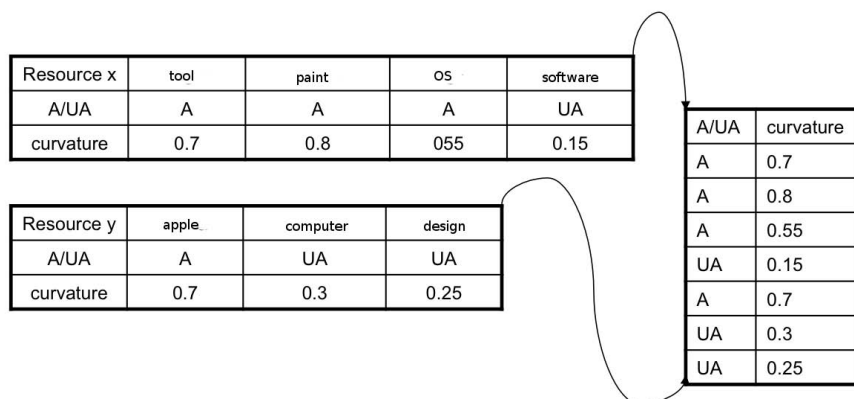


Fig. 2. Finding limiting values for tag (un)ambiguity

means that it is not clear whether a tag is ambiguous/unambiguous when the curvature value is in between the two limiting values. When this occurs during an automatic tag ambiguity calculation, feedback from the moderator is required.

Step 4: Testing. Because the calculation of the limiting values as well as the percentage of ambiguous tags per resource is based upon a small sample, there is no certainty that these numbers are well representing all the annotated resources from the community. We believe further testing is necessary before a real automatization of this process becomes possible. Therefore a new sample of randomly annotated resources has to be evaluated automatically as manually (by the moderators). In case 85% of the resources that are indicated as appropriately annotated by the approach correspond to the ones that are manually evaluated by the moderators, the next phase of the approach can start.

2.2 Community Runtime Phase

The approach can now be used to automatically select the appropriately annotated resources of the whole community. However, we believe that a reevaluation is imperative every six months since a community is always evolving over time. This means that previous steps have to be repeated every six months.

2.3 Community Information Retrieval Phase

Now, we can embed the results, appropriately annotated resources, into the search functionality. Each time the user gives a search task at the engine, the relevance of the results can be calculated based on previous obtained results.

Later, the use of these limiting values can be used to quality limiting values for meta data. Here, we make could make use of these limiting values to stimulate a content producer: in case the produced tags do not pass the limiting values, the users could be kindly asked (but not blocked) to improve his or her tags (e.g., by making it more specific). In a similar manner, content consumers are asked to detail tags that do not reach the quality limiting values.

3 Case Study

The approach presented in this paper is tested on case study involving a narrow folksonomy, i.e. a research group's blog. We tested the first phase only at this point. On this blog registered members can produce and consume content. However only the creators of the posts can annotate their content with tags. Because it is a closed environment there are not so many registered users.

3.1 Community Bootstrap Phase

We asked three members of the community to become a moderator. Selection was done based on their contribution to the system. A user who produces a large amount of data can be considered as someone who has read all the content which is produced and discussed on the system and is therefore a appropriate candidate for being a moderator. Up to the moment of writing, 1.550 postings have been created. We randomly selected posts, which share a tag that is frequently used by looking at the community's tag cloud. The larger the text font of a certain tag the more often used. The moderators were asked to evaluate whether the resources are appropriately annotated as defined in Sect. 4. We found that:

- general posts have only a small number of tags, ranging from 2 to 7. The system is set so that at least one tag is required;
- the community moderators sometimes had additional tags they would attach to the post, even in the case that the posts were originally their own as well as the tags;
- the community moderators deemed less than 25% of the tags for a post ambiguous, based on their understanding of the real-world domains involved in the research lab. The unclarity resulted from several reasons, most notably terms that were too general (e.g., the tag context), specific terminology (e.g., the tag frisco as in the the FRISCO report on information systems [4]), and obscure abbreviations (e.g., the tag VE for Visual Editor).

4 Discussion and Conclusion

In this paper, we presented a three-phase iterative approach to discover the properties of tags that are *appropriately annotated* to resources in a narrow folksonomy (e.g. pictures on Flickr) in order to improve information retrieval. We defined a resource as *appropriately annotated* when (1) the tags are well reflecting the content of the resource, (2) tags are expressed in such a way that they can easily be understood by the members of the community and (3) the resources can easily be found through tag search.

In our approach, we suggested the calculation of tag ambiguity, based on the curvature of tags in a graph, as well as the introduction of a community moderator, to deduce the properties of resources that are *appropriately annotated*. In the first phase, the community bootstrap phase, an approach to calculate the properties was proposed and in the second phase the resources were automatically selected based on the properties obtained in previous phase.

We tested the first phase (community bootstrap) of the approach in a case study involving a narrow folksonomy (a blog of research group). First results show that resources are not always *appropriately annotated* since in some cases there are tags that can be added to the resource. We believe it would be interesting to transform a narrow folksonomy into a broad folksonomy to let a resource annotate by several people instead of merely one person. About 25% of the tags appropriately annotated to a resource in the experiment were ambiguous.

We plan extended experiments on the first phase as well as on the other phases of our approach (community run time and community information retrieval) as future work.

References

1. Al-Khalifa, H., Davis, H.: Towards better understanding of folksonomic patterns. In: Proceedings of the 18th conference on Hypertext and hypermedia, pp. 163–166. ACM Press, New York (2007)
2. Damme, C.V., Hepp, M., Coenen, T.: Quality metrics for tags of broad folksonomies. In: Proceedings of the third International Conference on Semantic Systems (I-Semantics), J.UCS (2008) (forthcoming)
3. Dorrow, B.: A Graph Model for Words and their Meanings. Phd thesis, University of Stuttgart (Faculty of philosophy and history), Germany (2006)
4. Falkenberg, E.D.: Frisco: A framework of information system concepts. Technical report, IFIP WG 8.1 Task Group (1998)
5. Golder, S., Huberman, B.: Usage patterns of collaborative tagging systems. *Journal of Information Science* 32(2), 198–208 (2006)
6. Guy, M., Tonkin, E.: Tidying up tags. *D-Lib Magazine* (2006)
7. Lee, S., Han, S.: Qtag: introducing the qualitative tagging system. In: Proceedings of the 18th conference on Hypertext and hypermedia, pp. 35–36. ACM Press, New York (2007)
8. Moor, A.D., Leenheer, P.D., Meersman, R., Starlab, V.: Dogma-mess: A meaning evolution support system for interorganizational ontology engineering. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) ICCS 2006. LNCS (LNAI), vol. 4068, pp. 189–203. Springer, Heidelberg (2006)
9. Quintarelli, E.: Folksonomies: power to the people (2005)
10. Schmitz, P.: Inducing ontology from flickr tags. In: Proceedings of Collaborative Web Tagging Workshop at WWW 2006, Edinburgh, UK (2006)
11. Sen, S., Harper, F., LaPitz, A., Riedl, J.: The quest for quality tags. In: Proceedings of the 2007 international ACM conference on Conference on supporting group work, pp. 361–370. ACM, New York (2007)
12. Sen, S., Lam, S., Cosley, D., Frankowski, D., Osterhouse, J., Harper, F., Riedl, J.: Tagging, communities, vocabulary, evolution. In: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work, pp. 181–190. ACM, New York (2006)
13. Specia, L., Motta, E.: Integrating folksonomies with the semantic web. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 624–639. Springer, Heidelberg (2007)

EI2N 2008 PC Co-chairs' Message

It is a fact that enterprises need to collaborate in order to prosper in the current extremely dynamic and heterogeneous business environment. Enterprise integration, interoperability and networking are the major disciplines focusing on how to enable companies to collaborate and communicate in the most effective way. These disciplines are well-established and are supported by international conferences, initiatives, groups, task forces and European projects, where different domains of knowledge have been considered from different points of view and considering a variety of objectives (e.g., technological or managerial).

Enterprise integration involves breaking down organizational barriers to improve synergy within the enterprise so that business goals are achieved in a more productive and efficient way. Enterprise modelling, architecture, and ontology are the pillars supporting the achievement of enterprise integration and interoperability.

After the successful second edition of the workshop, in 2006, the third edition of the Enterprise Integration, Interoperability and Networking workshop (EI2N 2008) was organized as part of the OTM 2008 Federated Conferences and was sponsored by the IFAC Technical Committee 5.3 Enterprise Integration and Networking, the IFIP Working Group 5.12 Architectures for Enterprise Integration, and the European Virtual Laboratory on Enterprise Interoperability. The workshop aims to identify current research and practical issues on interoperability for enterprise integration that should be fully developed in the future. As a result of peer reviews, 8 papers were accepted out of 16 submissions. In addition to the presentations of the accepted papers, groups were organized to discuss the topics addressed, in order to involve the participants and gauge the impact of the workshop. These groups finally reported the results of the respective discussions.

The papers published in this volume of proceedings contribute to the domain of enterprise integration, interoperability and networking by presenting current research in the enterprise modelling, systems interoperability, services orchestration, and, more globally, systems engineering domains. While enterprise modelling is a prerequisite for the analysis of enterprise processes, process reference models and model-driven engineering are crucial to define business-IT alignment for networked enterprises, taking into account the interoperability issues induced by heterogeneous systems. However, they could take advantage of service-oriented architecture (SOA) technology, either within a single enterprise or among networked collaborative enterprises. The quality of these enterprise models has to be evaluated through maturity metrics before engineering the interoperability characteristics of the enterprise applications involved in the product value chain.

It has been a great pleasure to work with the members of the international program committee, who dedicated their valuable time and energy to reviewing, in time, the submitted papers; we are indebted to all of them.

We also would like to thank all authors for their contribution to the workshop objectives and discussions.

November 2008

Hervé Panetto
Arturo Molina
Peter Bernus
Andrew Kusiak

Business-IT Alignment Domains and Principles for Networked Organizations: A Qualitative Multiple Case Study

Roberto Santana Tapia^{1,*}, Maya Daneva¹, Pascal van Eck¹,
Nicté-Há Castro Cárdenas², and Leida van Oene³

¹ Department of Computer Science, University of Twente,
P.O. Box 217, 7500 AE Enschede, The Netherlands
{r.santanatapia,m.daneva,p.vaneck}@utwente.nl

² Department of Information Management, Government of the State of Tamaulipas,
16 Juárez, C.P. 87000 Ciudad Victoria, Tamps. Mexico
nichte.castro@tamaulipas.gob.mx

³ Department of Financial and Information Resources, Province Overijssel,
P.O. Box 10078, 8000 GB Zwolle, The Netherlands
A.v.Oene@overijssel.nl

Abstract. Applying principles for business-IT alignment in networked organizations seems to be key for their survival in competitive environments. In this paper, we present a qualitative multiple case study conducted in three collaborative networked organizations: (i) an outsourcing relation between an international IT and business integrator and a mass-marketed service provider, (ii) an inter-organizational collaboration among governmental departments of the state of Tamaulipas in Mexico, and (iii) a networked organization between the province Overijssel, the municipalities Zwolle and Enschede, the water board district Regge & Dinkel and Royal Grolsch N.V. in the Netherlands. Drawing from this case study, we derive four principles that networked organizations seem to adhere to when striving for alignment at a certain level of maturity.

Keywords: Alignment principles, inter-organizational cooperation, business networks.

1 Introduction

Despite years of research, aligning IT solutions with business needs remains one of the modern-day areas of concern for both business practitioners and researchers. Interest in business-IT alignment (B-ITa) is stimulated by cases of organizations that have successfully aligned their IT to gain competitive advantage and to improve organizational performance [1].

* Supported by the Netherlands Organization for Scientific Research (NWO) under contract number 638.003.407 (Value-Based Business-IT Alignment).

Several research studies claim that B-ITa can be achieved at various levels of maturity. Therefore, maturity models (MMs) seem a suitable vehicle for organizations to use in order to gain a deeper understanding of how they progress toward better alignment. Although we can find MMs to assess B-ITa (e.g., [23,4]), to the best of our knowledge there is no MM that specifically addresses the aspects needed for achieving alignment between business and IT in collaborative networked organizations (CNOs). CNOs arise when organizations redesign their structure to cooperate with other enterprises to address increasing competitive pressure in their markets. In our research, we are developing a MM to assess B-ITa in CNOs: the ICoNOs MM¹. We believe that achieving B-ITa in CNOs differs from achieving B-ITa in single organizations because in such settings, B-ITa is driven by economic processes instead of by centralized decision-making processes.

In earlier publications [5,6], we have reported on our motivation for developing the ICoNOs MM and on how we began to validate the model. In this paper, we describe a qualitative multiple case study conducted in three CNOs (two single case studies presented in our earlier work [6,7], and a new one). We used this multiple case study to identify the B-ITa domains included in the ICoNOs MM, i.e., partnering structure, IS architecture, process architecture and coordination. A domain is a group of processes that helps to have improvements in a particular CNO's area. The results of this study also led us to induce B-ITa theory in the form of principles that can be used in CNO settings when striving for B-ITa. The term 'principles' requires some explanation. Principles, in our context, are fundamental statements concerning B-ITa that should be helpful to CNOs. The principles should be helpful because they summarize important insights of B-ITa that we found in three real-life CNOs when striving for B-ITa. The principles crosscut B-ITa domains. The terms 'domains' and 'principles' are not used as synonyms in this paper.

The rest of this paper is organized as follows: in Sect. 2, we outline our theoretical framework. Section 3 describes our research approach explaining the research question, the rationale for the selection of the case study sites, the data gathering and analysis techniques, and the results. Then, Sect. 4 discusses the B-ITa principles for CNOs. Finally, Sect. 5 concludes the paper.

2 Definitions and Assumptions

2.1 Business-IT Alignment

B-ITa is, in this paper, defined as the *collaborative process to create an environment in which IT services support the requirements of the business*, whether such services are individually or collaboratively offered. We do not consider alignment as a steady state but as an operational process that needs to be improved continuously. With the term 'IT services' we mean the services offered by the information systems (software applications including the supporting infrastructure and

¹ The acronym ICoNOs MM stands for **IT-enabled Collaborative Networked Organizations Maturity Model**.

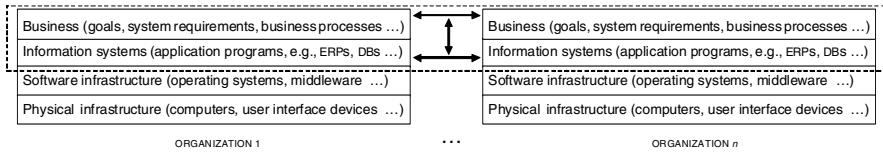


Fig. 1. Business-IT alignment framework

maintenance processes) to match the business requirements. By ‘requirements of the business’ we mean the systems requirements derived from analyzing the goal(s) of the CNO.

We analyze the B-ITa concept in CNOs based on the scheme shown in Fig. 1. The horizontal layers classify entities in a service provisioning hierarchy in a business: physical entities provide services to a software infrastructure, which provides services to information systems, which provide services to businesses. Participating organizations in a CNO need both to fit the different entities (horizontal arrows) as well as to address B-ITa (vertical arrow). Our interest is in the upper two layers of the framework (the area delimited by the dotted line), because there is where the business and IT alignment in CNOs takes place.

2.2 Collaborative Networked Organizations

We define a CNO to be any “mix-and-match” network of profit-and-loss responsible organizational units, or of independent organizations, connected by IT, that work together to jointly accomplish tasks, reach common goals and serve customers over a period of time [8]. Our interest is in IT-enabled CNOs, i.e., collaborations that are made possible by IT where the participants interoperate each other by means of information systems.

CNOs continue spreading since hypercompetitive environments [9] exist. Hypercompetitive environments force organizations to re-think the way they are doing business by connecting and aligning the business and IT operations among them to meet organizational goals. Participants in a CNO can be seen as distinct loosely coupled stakeholders with commonly conflicting interests and goals [10]. However, if they want to collaborate, they need to formulate a clear-enough common goal(s) toward which they strive together.

To thrive in their environments, which essentially are characterized by rapid changes in IT, easy competitors’ market entry and uncertain market demands, CNOs should be dynamic. Therefore, we consider that CNOs can change from moment to moment. Having well-defined collaborative work structures as basis, participants need to react promptly to customer needs [11,12]. They will collaborate while mutually interesting ‘business’ opportunity exists. When this opportunity is over, the CNO dissolves while, perhaps, the organizations are active in other CNOs or look for new ‘business’ opportunities.

2.3 The ICoNOs MM

Maturity models have been around for almost 15 years. A MM is an instrument that assesses specific domains against a norm. Based on maturity assessments, organizations know the extent to which activities in such domains are predictable. That is, organizations can be aware of whether a specific area is sufficiently refined and documented so that the activities in such area now have the potential to achieve its desired outcomes. The ICoNOs MM will help CNOs to assess the maturity of B-ITa activities to identify lacks of efficiency that can have a negative impact.

The ICoNOs MM is a two-dimensional framework. These dimensions are the maturity levels and the domains to which these levels apply [13]. In the following, we give a short summary of the domains included in the MM.

- Partnering structure, defined as the inter-organizational work division, organizational structure, and roles and responsibilities definition that indicate where and how the work gets done and who is involved.
- IS architecture, defined as the fundamental organization of the information management function of the participants embodied in the software applications that realize this function, their relationships to each other and to the environment, and the principles guiding its design and evolution.
- Process architecture, defined as the choreography of all (individual and collaborative) processes needed to reach the shared goals of the participants.
- Coordination, defined as the mechanisms to manage the interaction and work among the participating organizations taking into account the dependencies and the shared resources among the processes.

These domains have been identified in a literature survey and using the qualitative multiple case study presented in this paper. This identification was the main goal of our research (as presented in the next section). However, when analyzing the results of the case study, we recognized that some new statements could be derived. In this paper, we present these statements in the form of four principles that can be used by CNOs when striving for B-ITa (see Sect. 4). So, ‘domain’ and ‘principle’ are terms that should not be taken as synonyms. While domains refer to topics to consider when striving for B-ITa, the principles are general statements that crosscut domains and, therefore, are not applicable to specific domains only. The next section presents the research method we followed.

3 Research Approach

Multiple case studies enhance generalizability and reduce bias [14]. Therefore, this research uses a multiple case design. A multiple case study differs from a set of single case studies in that a multiple case study is an empirical inquiry that investigates a phenomenon within its real-life context to answer the same research question in different single case study sites. The objective of the study we conducted was to identify the necessary domains that CNOs must consider to

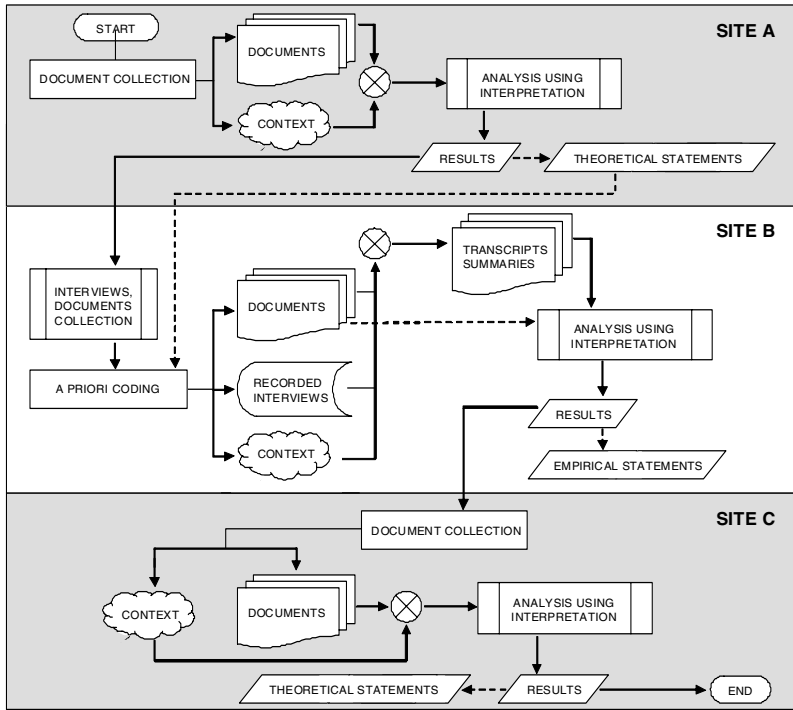


Fig. 2. Multiple case study method

achieve B-ITa. At the end, we found the four domains presented in the previous section. The case study findings also could be summarized in four B-ITa principles for CNOs (see next section). The research question to answer with this case was: *What are the necessary domains to consider when aligning IT with business needs in a CNO?*

A high-level view of our overall research process is presented in Fig. 2. It shows the way we conducted the study through the three case study sites. In the subsections that follow, we explain this in more detail.

3.1 Sites, Unit of Analysis and Timeline

Our main criterion for selecting the case study sites was the collaborative network perspective they take in the efforts for achieving B-ITa. B-ITa was the unit of analysis in this study. Once this criterion was met, the only other two requirements were that the CNOs explicitly had a B-ITa project and that they were willing to grant us access. To develop an overall MM we conducted our multiple case study in one entrepreneur-led CNO (site A) and two governmental CNOs (sites B and C). We did not intend to conduct a comparative study across CNOs, but rather to enrich our MM by bringing different insights from each CNO. In this sense, it is important to consider the particular context of each

CNO when interpreting the data (see Fig. 2). Therefore, we chose a hermeneutic approach [15] to analyze it. In our particular case, a hermeneutic approach helps to obtain results from analyzing the information sources, the sites, and their organizational contexts altogether. The data were collected during April 2006 and April 2008. A summary of the CNOs' background can be found in the appendix.

3.2 Data Collection and Analysis Techniques

The data collection technique used in each case study site was individually selected. This choice is motivated by the resources at our disposal. We considered the use of interviews for all sites. However, it turned out that we could only collect data through interviews in site B because professionals from sites A and C were not available. We only obtained documentation as source of evidence in these two sites. The documentation was carefully used and was not accepted as literal recording of information and events. Furthermore, in site B we used semi-structured interviews with an average duration of 1 hour per interview. The interviews were taped to help writing the transcripts which we used for analyzing. In this site, documents were supplementary sources of data.

The data analysis was conducted using interpretation [15]. We bear out this decision by the following statements: first, as we explained above, documentation was an important data source in this multiple case study. Documents are not simply containers of meanings. They are collectively produced, exchanged, and consumed. They summarize many decisions made by more than one person for a specific purpose. Documents represent specific circumstances including different insights. Therefore, the analysis of documents requires interpretation [16]. Second, in site B, professionals, i.e., people, were the primary data sources. In such situation, interpretation also is a suitable analysis technique. Generally, people develop and use their own understanding and observations of themselves and their environment. Therefore, it was expected that the interviewees attached their own meanings to their answers in the interviews. People interpret their world and we, as observing researchers, interpret their interpretations.

We followed a replication logic [14] when conducting this qualitative multiple case study. That is, after conducting the first single case study in site A, we uncovered significant findings (theoretical statements) that led us to conduct a second and a third single case study with replication as immediate research goal. Only with replication of findings, such findings could be robust for generalization [14]. We also took some steps to counterpart validity threats. As we were uncertain whether external conditions could produce different case study results, we articulated these conditions more explicitly identifying different case sites. We chose CNOs from different countries, one international and two of national nature, one entrepreneur-led and two government agencies, and one with a large amount of participants and two with only 2 or 3 participating organizations. We must also note that the B-ITa key drivers they have are different. The key drivers of the case study site A are to control costs and to manage

risk, while the B-ITa key drivers of sites B and C are to improve quality and to increase effectiveness. With all these different conditions in the sites included in our case study, we countered external validity. Construct validity was counterparted by data triangulation (i.e., use of multiple sources of evidence) and having our case study reports reviewed by peers and by professionals of the case study sites [14].

3.3 Case Study Findings

Fig. 3 presents a summary of the findings of our case study. Detailed information of the results of the single case study in site A can be found in [6]. The findings in this site suggest that the **coordination** and the **partnering structure** domains were the most important topics for the work between Outsourcer and Insourcer. Cost management also was a domain that strongly affected their B-ITa efforts. We used these findings to derive some theoretical statements that helped to codify themes for the second case study.

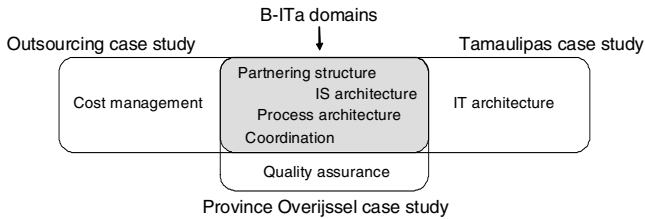


Fig. 3. B-ITa domains found in the validation case studies

As in site A, the findings of the case in site B suggested that **coordination** and **partnering structure** are indeed necessary domains that CNOs take into account in their B-ITa efforts [7]. However, the government of the state of Tamaulipas also consider the **process architecture**, **IS architecture** and **IT architecture**, as domains to address when striving for B-ITa. Following recommendations by Lee and Baskerville [17], with these results we could derive empirical statements from the theoretical statements created in the previous case study. It helped us to generalize results. However, we decided to conduct a third case study.

The results of the case study in site C replicated our previous findings. This CNO considers the four B-ITa domains as key areas to work on. It must be noted that for them **partnering structure** and **process architecture** were the most important domains. Also, in this case, we could identify a domain that is distinctive from the B-ITa domains. It is **quality assurance**. They consider testing, verification and control as activities that need to be present in all projects. The quality team is always trying to assure quality in the B-ITa project we had access.

Despite of the variation in the findings of sites A, B and C, the ICoNOs MM includes only four B-ITa domains. We did not consider ‘cost management’, ‘IT architecture’ and ‘quality assurance’ as B-ITa domains because such domains are not replicated in the single cases.

4 B-ITa Principles

We used the results of our case study to develop new theory in the form of principles that can be used by CNOs when striving for B-ITa. Below, we present these principles along with a short discussion on them.

Principle 1 (B-ITa domains). *Partnering structure, IS architecture, process architecture and coordination are necessary B-ITa domains to consider in CNOs.*

Clearly, one might argue that there are more domains that must be addressed when striving for B-ITa. However, we were looking for the necessary domains within the entire population of those domains that are sufficient to achieve B-ITa in CNOs. The necessary domains are the minimal number of domains that must be taken into account to achieve B-ITa in CNOs. The rest of the domains are additional domains that might be considered in B-ITa improvement attempts, but are not necessary. For example, an additional condition for achieving B-ITa in CNOs would be cost management (as found in site A). But while managing costs could be important, it is by no means necessary for achieving B-ITa.

Principle 2 (Domains order). *The order in which the domains are taken into account by CNOs in their efforts to achieve B-ITa should not affect the results.*

In the case study, we found that the importance of the domains varies according to the settings where a CNO works. When aligning IT with the business, each CNO can work in the domains that best meet its objectives. As our model is a continuous MM [18], it will let CNOs focus, for instance, on the domains with a low level of maturity. Those domains that are associated with higher maturity can, then, be included in later improvement efforts. For example, in site B, the government of Tamaulipas left the partnering structure unchanged. They concentrated in process architecture, IS architecture, and coordination in its B-ITa effort.

Principle 3 (B-ITa approach). *A top-down approach, which starts from strategic goals and plans to business/IT activities, should ensure B-ITa and provide value.*

Although we found that the order to address the B-ITa domains was not a considerable shortcoming to achieve alignment, we believe that an approach to strive for B-ITa is needed to provide ways to measure value and make real B-ITa improvements. Figure 4 presents a B-ITa approach. This approach is not the only one but it could work in practice since we derived it from the case study findings and it covers steps presented in other B-ITa methods (e.g. [19,20]). It addresses the four B-ITa domains presenting a view from an organization-centered perspective (i.e., the organization and structure of a CNO must exist a priori and the participants ought to follow it), where the first five steps of the approach are performed compulsorily before the rest.

Using a different perspective (e.g., a process-centered perspective), the relations (i.e., the arrows) between steps in Fig. 4 can be differently presented. Our approach does not contradict **Principle 2** but it does structure CNO B-ITa

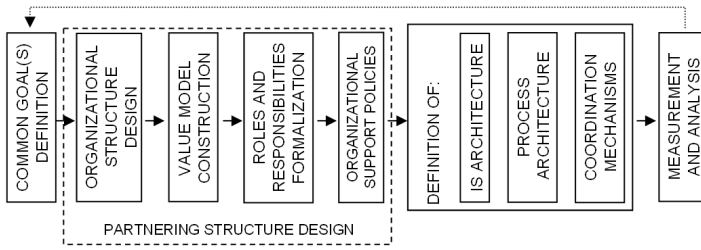


Fig. 4. B-ITa approach: A high level view

efforts from an organizational view. Explanation of most of the steps can be found in our earlier work [12]. The measurement and analysis step helps to define measures (e.g., financial and internal) and communicate the B-ITa results in order to consider them in future B-ITa projects to ‘assure’ quality and real improvements.

Principle 4 (B-ITa strategy). *Swift-reacting but in a delayed form is a strategy to follow when achieving business-IT alignment in networked organizations.*

As CNOs are dynamic, they need to react quickly to customers needs and to run apace B-ITa projects in order to survive in a hypercompetitive environment. For example, sites A and C are almost constantly analyzing their present IS architecture and how it helps them in order to align their strategies to the present situations. They also use, for instance, customer satisfaction analysis and market studies to define B-ITa projects. They react fast, but before ‘doing’, they take considerable time in discussing, planning and analyzing such projects. Our findings in this sites replicate the results presented by de Koning and van der Marck where they show how “no case, no go” [3, p.45] (i.e., if there is no positive fact-based explanation of the reasons for investing in IT, no action should be taken) is the motto of the most succesful business in the Netherlands for B-ITa-related decisions. It is to avoid unfavorable future results.

According to empirical research methodologies, we can conclude these four principles are empirically valid since we have justified them through the evidence provided by the multiple case study we conducted. However, we still consider them as hypotheses that can be admitted as true general principles only when they can, under other circumstances, be deduced again from verifiable observations.

5 Conclusion

The multiple case study presented in this paper has contributed to identify, and by means of replication, to validate necessary domains that should be considered by collaborative networked organizations when striving for business-IT alignment. The study has helped us to continue with the development of a maturity model to assess this alignment in collaborative settings. However, when

analyzing the data, we were able to propose four principles that might help networked organizations in their efforts for aligning IT services with business requirements. These principles relate to the domains included in our model, the order to address them, and an approach and strategy to achieve alignment.

Future work includes justifying these principles on the basis of evidence provided by other cases. Only after that, these principles can be considered fundamental statements that must be taken into account for alignment improvement efforts in collaborative settings. The principles presented in this paper are open to further empirical confirmation or refutation. Although much more research is required on this important topic for networked organizations, we hope that our study will contribute to the growth of this relevant research stream.

References

1. Sabherwal, R., Chan, Y.: Alignment between business and IS strategies: A study of prospectors, analyzers and defenders. *IS Research* 12, 11–33 (2001)
2. Federal Arch. Working Group: Architecture alignment and assessmentguide (2000)
3. de Koning, D., van der Marck, P.: *IT Zonder Hoofdpijn: Een Leidraad voor het Verbeteren van de Bedrijfsprestaties*. Prentice-Hall, Englewood Cliffs (2002) (In Dutch)
4. Luftman, J.: Assessing IT-business alignment. *Inf. Syst. Mngmt.* 20, 9–15 (2003)
5. Santana Tapia, R.: A value-based maturity model for IT alignment in networked businesses. In: Dubois, E., Pohl, K. (eds.) *CAiSE 2006*. LNCS, vol. 4001, pp. 1201–1208. Springer, Heidelberg (2006)
6. Santana Tapia, R., Daneva, M., van Eck, P.: Validating adequacy and suitability of business-IT alignment criteria in an inter-enterprise maturity model. In: *Proceedings of the Eleventh IEEE International EDOC Enterprise Computing Conference*, Annapolis, MD, USA, pp. 12–213. IEEE Computer Society Press, Los Alamitos (2007)
7. Santana Tapia, R., van Eck, P., Daneva, M.: Validating the domains of an inter-organizational business-IT alignment assessment instrument. Technical Report TR-CTIT-08-53, University of Twente, Enschede, The Netherlands (2008)
8. Santana Tapia, R.: What is a networked business? Technical Report TR-CTIT-06-23a, University of Twente, Enschede, The Netherlands (2006)
9. Bogner, W., Barr, P.: Making sense in hypercompetitive environments: A cognitive explanation for the persistence of high velocity competition. *Organization Science* 11(2), 212–226 (2000)
10. Damian, D.: Stakeholders in global requirements engineering: Lessons learned from practice. *IEEE Software* 24(2), 21–27 (2007)
11. Camarinha-Matos, L.M., Afsarmanesh, H.: A modelling framework for collaborative networked organizations. In: Camarinha-Matos, L.M., Afsarmanesh, H., Ollus, M. (eds.) *Network-Centric Collaborations and Supporting Frameworks*. IFIP Int. Federation for Information Processing, vol. 224, pp. 3–14. Springer, Boston (2006)
12. Santana Tapia, R., Zarvić, N.: Value-based partnering structure design for networked businesses: A multi-method approach. In: *Proceedings of 21st Bled Conference "eCollaboration"*, Bled, Slovenia, pp. 263–276 (2008)
13. Santana Tapia, R., Daneva, M., van Eck, P., Wieringa, R.: Towards a business-IT alignment maturity model for collaborative networked organizations. In: *International Workshop on Enterprise Interoperability – IWEI* (accepted, 2008)

14. Yin, R.K.: Case study research: Design and methods, 3rd edn. Applied Social Research Methods Series, vol. 5. Sage Publications, Thousand Oaks (2003)
15. Klein, H., Myers, M.: A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly* 23(1), 67–93 (1999)
16. Finnegan, R.: Using documents. In: Sapsford, R., Jupp, V. (eds.) *Data Collection and Analysis*, pp. 138–151. Sage Publications, Thousand Oaks (1996)
17. Lee, A., Baskerville, R.: Generalizing generalizability in information systems research. *Information Systems Research* 14(3), 221–243 (2003)
18. Santana Tapia, R., Daneva, M., van Eck, P.: Developing an inter-enterprise alignment maturity model: Research challenges and solutions. In: Rolland, C., Pastor, O., Cavarero, J.L. (eds.) *Proc. of the 1st Int. Conf. on Research Challenges on Information Science (RCIS 2007)*, Ouarzazate, Morocco, pp. 51–59 (2007)
19. Weill, P., Ross, J.W.: *IT Governance: How top performers manage IT decisions rights for superior results*. Harvard Business School Press (2004)
20. Chen, H.M., Kazman, R., Garg, A.: BITAM: An engineering-principled method for managing misalignments between business and IT architectures. *Sci. Comput. Program.* 57(1), 5–26 (2005)

Appendix: Background of the CNOs studied

Site A: Technology Outsourcing Relation.
<p>The first CNO we studied was an outsourcing relationship between a leading international business and technology integrator and a local provider of mass-marketed services (hereinafter referred to as: Insourcer and Outsourcer, respectively).</p> <p>Insourcer is an international IT services company, providing consultancy, systems integration and managed operations. Outsourcer offers a wide range of services for both the private and business market. The stock of both Insourcer and Outsourcer is traded on Euronext, the pan-European stock exchange, where both companies are in the top 100 in terms of market capitalization. Outsourcer decided to outsource part of its IT operations to Insourcer in 2001 and this was a measure to confront the company’s troubled IS architecture management. The 2001 architecture consisted of home-grown applications, stove pipe solutions, and a lot of point-to-point connections. The company experienced problems related to inconsistent data, significant operational expenses, and below-average customer satisfaction. The outsourcing measure had the objective to help (i) provide continuity of service to the customers so that number of complaints is reduced, (ii) improve financial results due to purchase price and cost reduction, and (iii) optimize the IT architecture and performance.</p>
Site B: State Government Collaboration.
<p>A network of more than hundred departments of the state of Tamaulipas in Mexico was the second CNO we studied. As a response to the necessity of having a modern government administration, the government of Tamaulipas implemented Domino/Notes to allow the departments to maintain fast and uninterrupted internal communication. Their overall requirements were to make the service-delivery process more effective and efficient, and to create a better government-citizen relation responding to the society expectations.</p> <p>The first project under Domino/Notes was the Citizen Attention Service System (CASS). This system helps to collect all the individual requests and petitions that the citizens raise to the government. The CASS project began in 2001. The initial situation in the area of service provisioning to citizens was characterized by much bureaucracy and poor response time. Only few of the departments had a system to manage the requests. Those systems were home-grown applications developed by IT sections of different departments. Each had its own application logic and data semantics and contributed in a unique way to a lack of homogeneity and communication among systems. The CASS facilitates the allocation, distribution and communication of citizens’ requests among departments, as well as all the information related to such requests. This helps to have better control in each of the processes, while having a close relation with the citizens to keep them informed on their requests’ process.</p>

Site C: Regional Government Network.

People who want to build, re-build, or re-use a house, factory, or barn, in the Netherlands, can often need to apply for licenses and permits regarding residency, spatial planning, and the environment. Each of these licenses and permits has their own set of criteria, procedures, administrative desks, waiting periods, fees, and staff. For both citizens and companies, this is a complex and time consuming process that costs both applicants and the government a great deal of money. The Ministry for Housing, Spatial Planning and Environmental Management (VROM – initials in Dutch) wants to gather the different licenses together within the 'omgevingsvergunning' – the environmental permit. All aspects can then be requested from a single point of contact to obtain a decision although such decision needs the collaborations of different organizations.

The environmental permit project is part of a set of measures that has been initiated to substantially reduce administrative charges for citizens and businesses. From January 1st 2009, municipalities, provinces and water board districts should be able to use the new process. The environmental permit is part of the modernization plan for VROM legislation, in which the ministry is reducing and improving its rules and regulations. The project includes a development of an implementation plan with pilot projects and advice. The third CNO we studied was one of this pilot projects. It is a networked organization among the province Overijssel, the municipalities Zwolle and Enschede, the water board district Regge & Dinkel and Royal Grolsch N.V.

The View-Constraint Duality in Database Systems, Software Engineering, and Systems Engineering

John A. Springer¹ and Edward L. Robertson²

¹ Department of Computer and Information Technology
Purdue University

West Lafayette, IN 47907-1421
jaspring@purdue.edu

² Computer Science Department
Indiana University

Bloomington IN 47405-4101
edrbtn@indiana.edu

Abstract. In database systems, software engineering, and systems engineering, the concepts of constraints and views are commonly and effectively used. Considered distinct, they stand as well-established notions in each domain's body of knowledge. The focus of this paper is to explore the duality between views and constraints in these domains and investigate the efficacy of this duality in enabling more effective model interoperability. We provide empirical evidence for the duality and demonstrate cases where the duality is useful for constraint specification across modeling paradigms as commonly occurs across multiple organizations.

1 Introduction

In database systems, software engineering, and systems engineering, constraints and views are commonly and effectively-used concepts. Considered as highly distinct, they stand as well-established notions in each domain's body of knowledge. In fact, they are duals, in that each is expressible in terms of the other. In the realm of systems analysis and modeling, the ability to specify constraints in terms of views is particularly useful; this ability will be the focus of this paper.

As currently deployed, constraints may be expressed as “business rules” independent of underlying models – one notable exception being cardinality constraints in Entity-Relationship (ER) models. Moreover, constraints generally have a local flavor, in that they are defined and evaluated with respect to a local, more limited context. In general, the modeler and designer may be unable to fully and precisely express constraints where they arise, so they are “baked” into the information system. That is, the modeler or designer may employ a model or methodology that does not support natively the specification of a particular constraint, where the constraint itself is most naturally understood in the context of the model/methodology. For example, consider the case where a functional constraint (*i.e.*, business rule) applies most naturally to the entities in an ER model, but the conventional ER notation does not support the specification

¹ For example, in the database realm, the tuple/table distinction is the local/global one.

of such a constraint. As a result, such constraints only manifest themselves in the information system, and consequently, one is only aware when the constraint has been violated, in effect recording the failure rather than catching the failure. We denote instances where constraints do not have native support as cases of *constraint specification displacement*.

As for views, we may informally define a view as an organized subset of a whole, where the subset is chosen according to some criteria. The term “view” usually denotes an external representation meant to conceal the internal structure of the whole and display only the information pertinent to the intended viewer. Moreover, a view typically has a global nature (that is, one usually defines them in a global manner). By virtue of this, views often have a distinctly different flavor than that possessed by constraints.

The focus of this paper is to explore the duality between views and constraints in these domains and investigate the efficacy of this duality in enabling more effective model interoperability. We provide empirical evidence for the duality and demonstrate cases where the duality is useful for enabling constraint specification across modeling paradigms as commonly occurs across multiple organizations. The accuracy with which the duality may be demonstrated depends upon the degree to which the two concepts can be formalized; in some cases, positing the duality actually aids in formalizing one of the concept in terms of the other. In the case of the relational data model, the VIEW operation is well-defined, and there exist many formalizations of constraints that use this operation. Thus, the database domain readily exhibits the duality. In the case of the other models, constraints have a less formal expression, and views prove to be a method for interjecting more formality into the expression of the constraints.

Section 2 discusses the necessary background information and introduces the notational conventions that we employ. Section 3 begins the process of finding empirical evidence of the duality by closely inspecting the database systems domain, continued in Section 4 in the other domains. In Section 5 the value of the duality in enabling model interoperability is demonstrated. Section 6 summarizes our results and discusses future work.

2 Background

This section provides background material and related work pertaining to the view-constraint duality. We start with the notational conventions employed in the remainder of the document. Other conventions of more limited scope are introduced as necessary.

- $R(A_1, \dots, A_n)$ and $S(B_1, \dots, B_m)$ are relation schemas.
- \mathbf{r} is an instance of R , and \mathbf{s} is an instance of S .
- t_1 and t_2 denote tuples.
- α denotes transitive closure.

We use the Zachman Framework for Enterprise Architecture [1] to help categorize the contexts for views and constraints. The relational model, relational query languages, the relational model’s dependencies and constraints, and SQL VIEWS are covered in [2]. A well understood property of SQL is that it may serve as a constraint specification language. Garcia-Molina, Ullman, and Widom discuss this in detail in [3]. A more

extensive review of the literature concerning views and constraints is given in the longer version of this paper [4].

Standards organizations have given significant attention to views and constraints in their publications, albeit as independent concepts. For instance, the Institute of Electrical and Electronics Engineers (IEEE) have formally defined views in the software engineering context [5]. In addition, in [6], the International Organization for Standardization (ISO) addresses the relationship between views and constraints by acknowledging the role that views play in understanding models and their constraints.

3 Duality in Data Models

The natural way to use views to express constraints is via set containment or (in)equality between view expressions. In the database systems domain (*i.e.* the Zachman Framework’s “What” column), views are expressed in SQL or more concisely in relational algebra expressions. We explore this with one simple example involving the well-known foreign key constraint (many other examples using the core relational model are given in [7]) and one complex example that requires operations beyond the core relational model.

The classic definition of a foreign key constraint (Example 1) involves quantification and thus is less accessible to domain experts, while the version using views (Example 2) is conceptually simpler.

Example 1 (Foreign Key Constraint). A_i in \mathbf{r} is a foreign key that refers to B_j in \mathbf{s} provided that $\forall t_1 \in \mathbf{r}, \exists t_2 \in \mathbf{s} (t_1.A_i = t_2.B_j)$

Example 2 (Foreign Key Constraint Using Views). $\pi_{A_i}(\mathbf{r}) \subseteq \pi_{B_j}(\mathbf{s})$

While the foreign key example involves a direct translation from first-order logic (FOL), the constraint-via-views approach is even more effective in contexts when FOL is inadequate because the view expression can use operators beyond core relational algebra (operators that have no natural counterpart in FOL), including aggregation (γ , from [3]), transitive closure (α , from [8]), and relational metadata operations (\downarrow , Δ , and \mathcal{P} , from [9]).

For an example of an aggregation constraint, we use a scenario involving a fictional truck manufacturer wishing to insure that each vehicle has the correct number of specified parts, such as the correct number of tires for a particular type of truck. Figure 1 shows a (highly simplified) Entity-Relationship (ER) model that records information about vehicle types on the left and vehicle instances on the right. Figure 2 gives the corresponding database schema (cardinality constraints from the ER model are expressed separately).

Based on these tables, we can create an expression that reflects the constraint that a truck must have exactly 4 tires; in fact, we can express a more general constraint that restricts the part instances in the *Part* table to the quantities specified in the *SubpartS* table. Example 3 is this expression in relational algebra form. It is worth noting that since each part may have several *child* parts (*i.e.*, subparts) but only one *parent* part, the *Part* table contains the part/parent part combination instead of the more commonly seen

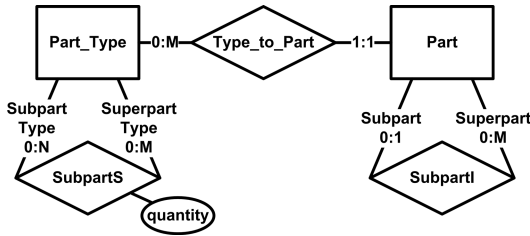


Fig. 1. Parts ER Diagram

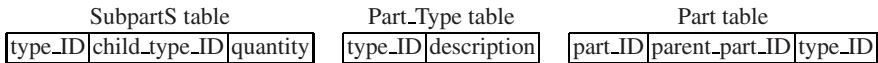


Fig. 2. Parts Database Schema

part/child part combination, and this accounts for the potentially unexpected grouping in Example 3.

Example 3 (Part Quantities Constraint). $\pi_{type_ID, subtype_ID, quantity}(PG) \subseteq SubpartS$ where PG represents the following expression:

$$Part \bowtie \gamma_{parent_part_ID \rightarrow part_ID, type_ID \rightarrow subtype_ID, COUNT(part_ID) \rightarrow quantity}(Part)$$

That constraints expressed in this form are already used only serves to enhance the need for according view-based constraints first-class treatment.

4 Duality in Other Models

In this section, we explore models outside the Zachman Framework’s What column. Such models include the process model and the time-based model. The process model resides in the How column, and the time-based model has its origin in the When column. In this paper, we limit our discussion to time-based models, although we also have results for process models. The view expressions that we exhibit often use operators or notions that are quite close to those of the extended relational algebra. Observing the principle that most production constraints are in fact enforced via the information system, this means that the translation from specification to implementation is direct and hence easy to verify.

Examples of models in the When column include data flow diagrams, in the tradition of [10], as well as UML state and sequence diagrams [11]. The master production schedule is the prototypical model from Zachman’s When-Specify [1] cell. As discussed in [12], a master production schedule is an outline of the products that will be made in each time period (e.g., a week) of a production plan. In addition, Portugal [13] indicates that the aggregate capacity plan (that is, the planned production capacities for product lines), the inventory levels, and the short-term demand forecast provide the inputs for the master production scheduling process. Table 1 shows a master production

Table 1. Master Production Schedule

PRODUCT TYPE	WEEK			
	1	2	3	...
T1	300	120	190	...
T2	450	260	310	...

Table 2. Daily Production Schedule

DATE_TIME	TASK	TYPE_ID
⋮	⋮	⋮
08/19/2006 8:44	Assemble Chassis	T2
08/19/2006 8:52	Assemble Chassis	T1
⋮	⋮	⋮
08/19/2006 9:15	Add Trim	T2
08/19/2006 9:17	Add Trim	T1
⋮	⋮	⋮
08/19/2006 10:43	Finalize Truck	T2
08/19/2006 10:48	Finalize Truck	T1
⋮	⋮	⋮
⋮	⋮	⋮

schedule for our fictional truck manufacturer; MPS denotes this table and $MPS(p,w)$ denotes the cell indexed by product number p and week w .

To meet the goals established in the master production schedule, we naturally need more detailed scheduling that addresses daily production. As discussed in [14], several steps follow the creation of the master production schedule in the general scheduling process before actual manufacturing occurs, and the process varies based on the methods used by the parties that are responsible for scheduling. After the requisite materials and capacity planning [14], we then arrive at a daily production schedule that specifies for a particular date and time when each task in the assembly of a particular truck occurs and also tracks the part type identifier for the truck involved in each task. In Table 2, we list a portion of a daily production schedule.

To track its progress toward its goals, we naturally want to correlate the daily production to a master production schedule and constrain its efforts to meet the planned output. An effective approach is to create views on the schedules and then use these views to specify the constraint. As the daily production schedule and the master production schedule have grossly different levels of detail, we need a mechanism to bridge this gap, and to borrow the concept of aggregation (in particular count) from the relational database realm.

The first view is simply MPS or a cell $MPS(p,w)$ thereof. As for the second view, we want to compare values in MPS to the appropriate information from the daily production schedules. Naturally, we need a week of the daily production schedules; more specifically, we need the schedules for all of the dates that fall in the second week of the master production schedule.

To accomplish this, we choose from the entire collection of daily production schedules the subset of daily production schedules that fall in a specified week and count the number of trucks produced, observing that the “Finalize Truck” task indicates when a truck has been produced. This results in a view *Weekly_Production*, which is a two dimensional array parameterized by $TYPE_ID$ and week number. The definition of *Weekly_Production*, seen in Example 4, is done in two steps for clarity.

Example 4 (Weekly Truck Production Instances View).

$$\begin{aligned}
 \text{Daily_Production}(T,d) &= \text{the number of occurrences of "Finalize Truck" for trucks} \\
 &\quad \text{of type T on day } d, \text{ and} \\
 \text{Weekly_Production}(T,w) &= \sum_{d \in \text{WEEK } w} \text{Daily_Production}(T, d)
 \end{aligned}$$

We now have the two views, both two dimensional arrays, whose equality expresses the desired constraint, as seen in Example 5.

Example 5 (Daily Truck Production Schedule Constraint). MPS = Weekly_Production.

Example 5 demonstrate the capacity of views to express constraints in the context of time-based models. In this context, we employ an aggregate view to calculate a number that we then use in a comparison with another number.

Regarding the other case (views expressed with constraints) of the duality in the time-based model context, we build on Example 5 and from it, we create a series of views. Example 5 specifies the contents of the first view, parameterized by vehicle type T and week number w. The view contains at most one pair T and w, and it only contains this pair if the constraint is met. Example 6 holds this view expression.

Example 6 (View Based on Daily Truck Production Schedule Constraint).

$$\{(T,j) \mid \text{MPS}[i,j] = \text{Weekly_Production}(T,j) \text{ and } T \in \{“T1”, “T2”\} \text{ and } 1 \leq j \leq n\}$$

Alone, the view in Example 6 is sufficient evidence that views can be expressed with the aid of constraints in the time-based context. However, we can use Example 6 in another expression that serves to indicate instances where the constraint fails; Example 7 contains this expression.

Example 7 (Violator Identifying View Based on Example 6).

$$\{(T,j) \mid \text{and } T \in \{“T1”, “T2”\} \text{ and } 1 \leq j \leq n \} [\text{MPS}[i,j] \neq \text{Weekly_Production}(T,j)]$$

If the constraint fails, Example 7 generates the violators; if the constraint holds, it generates an empty set. The capacity to include Example 6 in another view (Example 7) provides additional empirical evidence of the presence of the duality in the time-based context. Example 7 also showcases yet another example of the effectiveness of the duality in pinpointing constraint violators.

5 Model Interoperability

In this section, we examine the duality in the context of model interoperability, and in this context, we focus on the interactions between process and time-based models. As we did in previous sections, we begin with the use of views in expressing constraints; in this particular section, we employ process and time-based model views in time-based constraints. We then demonstrate the use of time-based constraints in the construction of views on process and time-based models.

We investigate instances of the duality by utilizing process and time-based model views in time-based constraints. We continue to employ our fictional example. In Fig. 3, we depict the truck assembly process and its five major subprocesses. In that figure, we delineate dependencies (that is, relations) with directed edges, as is the accepted convention and the standard to which we conform in this section; for example, the

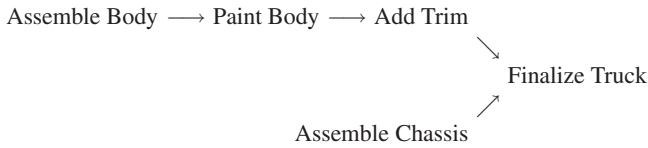


Fig. 3. Truck Assembly Process Flowchart

directed edge from the “Assemble Body” process to the “Paint Body” process indicates that the “Paint Body” process depends on the “Assemble Body” process. In addition, notice that the “Finalize Truck” process depends on the “Add Trim” and “Assemble Chassis” processes.

In addition to the requisite materials and capacity planning [14], we generate a model that describes the temporal sequence that must be obeyed during truck assembly. Known as an activity or task network diagram [15], it can range in detail from a high-level depiction such as the process model found in Fig. 3 to a diagram specifying the ordering of the atomic tasks in a particular process. For our purposes, a high-level depiction is suitable, and thus we employ Fig. 3 as the task network diagram. Henceforth, we refer to the processes in Fig. 3 as tasks.

The task network diagram impacts the daily production schedule in Table 2. In Table 3, we revise the Daily Production Schedule for August 19, 2006 (that is, Table 2) to include specific part identification numbers for the trucks; the tasks listed in Table 3 correspond to the tasks in Fig. 3.

A simple constraint on the daily production schedule is that for each product (that is, a truck), no task that follows another task in the task network diagram can precede that same task in the daily production schedule. For example, since the “Finalize Truck” task follows the “Add Trim” task in the task network diagram, it cannot precede the “Add Trim” task in the daily schedule. Obviously, this constraint does not have to hold for different products (e.g., trucks T4655 and T4656); in fact, given the nature of activities

Table 3. Revised Daily Production Schedule

Table 4. *Depends_On* relation

DATE_TIME	TASK	TYPE_ID	PART_ID
⋮	⋮	⋮	⋮
08/19/2006 8:44	Assemble Chassis	T2	T4656
08/19/2006 8:52	Assemble Chassis	T1	T4656
⋮	⋮	⋮	⋮
08/19/2006 9:15	Add Trim	T2	T4656
08/19/2006 9:17	Add Trim	T1	T4656
⋮	⋮	⋮	⋮
08/19/2006 10:43	Finalize Truck	T2	T4656
08/19/2006 10:48	Finalize Truck	T1	T4656
⋮	⋮	⋮	⋮

PREV_TASK	NEXT_TASK
Assemble Body	Paint Body
Paint Body	Add Trim
Add Trim	Finalize Truck
Assemble Chassis	Finalize Truck

on an assembly line, it is very likely optimal that the constraint does not hold. However, for an individual product, the constraint is mandatory.

The first step in using views to express this constraint involves the task network diagram. In this section, of primary importance is the order of the abstract tasks in the task network diagram (*i.e.*, the tasks specified in the model); the use of abstract tasks in this case is in sharp contrast to earlier in this section when our focus was on process instances. Toward that end, we create a *Depends_On* relation that captures the order of the abstract tasks. Table 4 provides the *Depends_On* relation for the tasks found in Fig. 3. Notice that the tuples in the *Depends_On* relation in Table 4 contain the abstract tasks, not instances. Furthermore, we use the *PREV_TASK* and *NEXT_TASK* attributes in the *Depends_On* relation to distinguish between the tasks involved in a particular dependency.

We next take the transitive closure of the *Depends_On*. Following the convention employed throughout this document, we represent transitive closure with the α operation; consequently, we express the transitive closure of *Depends_On* as $\alpha(\textit{Depends_On})$. Taking the transitive closure of this relation is necessary to create explicit dependencies between all of the tasks in the task network diagram.

We also utilize the daily production schedule from Table 3, representing it as the *DPS* relation with attributes *DATE_TIME*, *TASK*, *TYPE_ID*, and *PART_ID*. We then employ domain relational calculus notation to express a view that utilizes the *DPS* relation. This expression appears in Example 8, and we refer to the expression as *BEFORE*.

Example 8 (Task Succession View). $BEFORE = \{(a_p, a_s) \mid (d_p, a_p, i_p, p_p) \in DPS \wedge (d_s, a_s, i_s, p_s) \in DPS \wedge p_p = p_s \wedge d_p < d_s\}$

We are now prepared to express the constraint. It appears in Example 9 and uses the expression in Example 8 as well as $\alpha(\textit{Depends_On})$.

Example 9 (Task Succession Constraint). $BEFORE \subseteq \alpha(\textit{Depends_On})$

Simply stated, the constraint specified in Example 9 aims to eliminate the case where a task in the schedule disobeys the task network diagram. Moreover, Example 9 illustrates the capacity of views to express constraints in yet another context, the interoperability of models. We next address the other aspect of the duality – the ability to express views with constraints – in this same context.

We endeavor to discover instances of the duality by utilizing time-based constraints in the construction of views on process and time-based models; an instance of this quickly follows from the example in the previous subsection. Let us begin with a re-statement of Example 9 as expressed in Example 10.

Example 10 (Task Succession Constraint Using Empty Set). $BEFORE - \alpha(\textit{Depends_On}) = \emptyset$

We next repeat our actions from earlier and remove the empty set equality. The resulting expression is a view that pinpointing the pairs of tasks that do not obey the task network diagram. We state this view in Example 11.

Example 11 (View Based on Task Succession Constraint). $BEFORE - \alpha(\textit{Depends_On})$

Example 11 serves as a clear illustration of an interoperable modeling situation where one uses a constraint in the construction of a view. However, it has limited utility as a mechanism to identify the constraint violators; this limitation stems from its inability to determine to which truck a pair of tasks belongs. In turn, this inability results from the abstract nature of the *Depends_On* relation. That is, the *Depends_On* relation has as its domain the abstract tasks of Fig. 3 and an abstract task does not pertain to a specific truck. Consequently, the *Depends_On* relation has no notion of pairs of tasks for a *specific* truck (e.g., truck T4655); it simply contains the abstract dependencies.

Because of the nature of the *Depends_On* relation and the manner in which we specify the constraint (Example 9), the *BEFORE* view only contains pairs of tasks, and consequently, the information that matches tasks with trucks is lost. This motivates us to find another method for expressing this constraint, and we offer an alternative in Example 13. This example utilizes the *EARLY* view from Example 12.

Example 12 (Violator Identifying View Based on Task Succession Instances). $EARLY = \{S \mid S \in DPS \wedge \exists P \in DPS, \exists D \in Depends_On (P.PART_ID = S.PART_ID \wedge P.TASK = D.PREV_TASK \wedge S.TASK = D.NEXT_TASK \wedge P.DATE_TIME > S.DATE_TIME)\}$

Example 13 (Task Succession Instances Constraint). $EARLY = \emptyset$

In the *EARLY* view, we utilize tuple relational calculus to identify the set of tuples from *DPS* (i.e., tasks in the daily production schedule) that disobey the constraint; adding the empty set equality in Example 13 seeks to enforce the constraint. Moreover, we overcome the limitations of Example 9 by using the tuple variable *S* that contains date/time, task, type identifier, and part identifier information; consequently, we are able to identify the actual truck-task pairs that violate the constraint.

6 Conclusion

Throughout this paper, we observe that a duality does indeed exist between views and constraints in the database systems, software engineering, and systems engineering domains. Employing empirical evidence, our investigation reveals that the accuracy with which the duality holds depends upon the degree to which the constraints can be formalized. In the case of the relational data model, its canonical constraints are easily formalized. This is even true of the semantic constraints that are expressible in relational algebra and its extensions. In the case of the other models that we explore, the constraints have a less formal expression, and views prove to be a method for interjecting more formality into the expression of the constraints.

We also demonstrate that the duality enables model interoperability and thus empowers one to specify constraints in a natural way that minimizes *constraint specification displacement*. To fully realize the significance and capabilities of the duality as it concerns model interoperability, continued effort is needed to complement the results presented in this paper with tools that incorporate the duality, and this is the focus of our future work.

References

1. Zachman, J.A.: A framework for information systems architecture. *IBM Systems Journal* 26(3) (1987); IBM Publication G321-5298
2. Abiteboul, S., Hull, R., Vianu, V.: *Foundations of Databases*. Addison-Wesley, New York (1995)
3. Garcia-Molina, H., Ullman, J.D., Widom, J.: *Database Systems: The Complete Book*. Prentice Hall, Upper Saddle River (2002)
4. Springer, J.A., Robertson, E.L.: The view-constraint duality in database systems, software engineering, and systems engineering. Technical Report TBD, Computer Science Department, Indiana University (July 2008)
5. IEEE: IEEE Std 1471-2000, IEEE Recommended Practice for Architectural Descriptions of Software Intensive Systems. Institute of Electrical and Electronics Engineers, New York, USA (2000)
6. ISO: ISO 14258: Industrial automation systems – Concepts and rules for enterprise models. International Organization for Standardization, Geneva, Switzerland (2000)
7. Springer, J.A.: *View-Constraint Duality in Databases and Systems Engineering*. PhD thesis, Indiana University, Computer Science Department (August 2007)
8. Immerman, N.: Languages that capture complexity classes. *SIAM Journal on Computing* 16(4), 760–778 (1987)
9. Wyss, C.M., Robertson, E.L.: Relational languages for metadata integration. *ACM Trans. Database Syst.* 30(2), 624–660 (2005)
10. Ward, P.T., Mellor, S.J.: *Structured Development for Real-Time Systems*. Prentice Hall Professional Technical Reference (1991)
11. Popkin Software: Popkin Enterprise Architecture Framework (visited 4/23/2006), <http://government.popkin.com/frameworks/zachmanframework.htm>
12. Portougal, V., Sundaram, D.: *Business Processes: Operational Solutions for SAP Implementation*. Idea Group Publishing (2005)
13. Portougal, V.: XXIII: ERP Implementation for Production Planning at EA Cakes Ltd. In: *Cases on Information Technology: Lessons Learned*, vol. 7, Idea Group Publishing (2006)
14. Russell, R., Taylor, B.W.: *Operations Management: Quality and Competitiveness in a Global Environment*, 5th edn. John Wiley & Sons, Inc., Chichester (2006)
15. Pressman, R.S.: *Software Engineering: A Practitioner's Approach*. McGraw-Hill Science/Engineering/Math. (2004)

Mining Reference Process Models and Their Configurations

Florian Gottschalk, Wil M.P. van der Aalst, and Monique H. Jansen-Vullers

Eindhoven University of Technology, The Netherlands
{f.gottschalk,w.m.p.v.d.aalst,m.h.jansen-vullers}@tue.nl

Abstract. Reference process models are templates for common processes run by many corporations. However, the individual needs among organizations on the execution of these processes usually vary. A process model can address these variations through control-flow choices. Thus, it can integrate the different process variants into one model. Through configuration parameters, a configurable reference models enables corporations to derive their individual process variant from such an integrated model. While this simplifies the adaptation process for the reference model user, the construction of a configurable model integrating several process variants is far more complex than the creation of a traditional reference model depicting a single best-practice variant. In this paper we therefore recommend the use of process mining techniques on log files of existing, well-running IT systems to help the reference model provider in creating such integrated process models. Afterwards, the same log files are used to derive suggestions for common configurations that can serve as starting points for individual configurations.

1 Introduction

Many supporting processes like, e.g., procurement or invoicing processes are organized similarly among companies. Reference process models depict such processes in a general manner, thus providing templates for individual implementations of these processes, and are available for various domains [8,11,16]. However, even supporting processes are rarely executed in exactly the same manner among companies. For that reason the corresponding reference models must be adapted to individual needs during the process implementation.

To support such an adaptation of reference models, suggestions for configurable process models have been made by various researchers (e.g. [3,10,13]). Configurable process models require that the reference process model provider combines different process variants into an integrated process model from which reference model users can then derive individual model variants by setting configuration parameters. In this way, each organization can derive the individual process model fitting their needs while this derived process model remains conform with the used reference model. That means, the adaptation process does not require a manual process modeling which always comes with the risk of error and which could thus jeopardize a process's executability.

While this simplifies the adaptation process for the reference model user, it is on the expense of the model provider who has to construct a more complex model integrating several process variants. As handling the complexity and avoiding errors is already difficult for traditional reference models [6], we focus in this paper on providing support for the creation of configurable process models.

Usually, the processes represented by a reference model are not new and innovative approaches, but rather derived from established process variants. That means, when a configurable reference model is built, various variants of the process are already in place in organizations. In traditional reference model creation approaches, process analysts are examining the processes in place, e.g. through expert and user interviews, and then compile the gained information “in their minds” to a best-practice reference model while they abstract from specific requirements of individual organizations. A configurable model, however, should already provide everything that is necessary to derive a model satisfying the specific requirements. For that reason, it needs to include these specific aspects of different process variants. Thus, an abstraction is not needed and the model should rather be based on what is really happening.

Extensive protocols about what has happened are usually widely available in today’s IT environments in the form of so-called log files. Data and process mining techniques have been developed to gain condensed information about the process behavior from such log files. In the following we will depict how such existing techniques can directly support the creation of a configurable reference model by using the log files from various IT systems.

Figure 1 provides an overview of the suggested approach. At first, the available log files from various systems must be prepared through filtering of irrelevant content and mapping of different naming conventions among different log files. This is subject of Section 2. Afterwards, process mining techniques can be used not only to create process models for individual systems but also to build process models which are valid for all the systems, i.e. which integrate the various process variants. We show this in Section 3. Section 4 depicts how such integrated models covering multiple variants of a process can be configured to derive the individual variants before the paper ends with some conclusions.

2 Pre-processing the Log Files

Many of today’s IT systems constantly write log files to record functions that are executed in the system, changes that are made to the system or its data, “system alive” status updates and so on. For example, most web-servers write an entry into a log file for each single requested page including information about the time of access, the IP address of the user, whether the access was successful, and maybe even a user name or submitted data. Due to today’s extensive use of information systems in all business areas, such log files containing information about the executed business processes are usually widely available.

In this paper we focus on deriving the control flow among the different activities of a business process from such log files. For this purpose many details of log

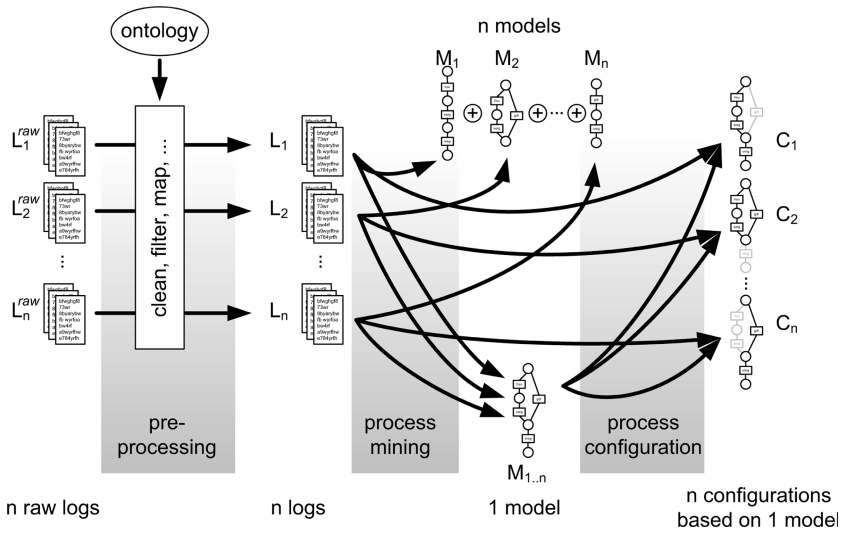


Fig. 1. Deriving integrated process models and their configurations from log files

files are irrelevant and it is here sufficient if we consider a log file as being a set of event traces. Each event trace is then an ordered set of the activity identifiers classifying each log event as the execution of the particular activity.

Definition 1 (Log file). Let E be a set of activity identifiers. Then

- $I = E^*$ is the set of all possible event traces, i.e. $\langle e_1, \dots, e_n \rangle \in I$
- $events : I \rightarrow \mathcal{P}(E)$ is a function defined such that $events(\langle e_1, \dots, e_n \rangle) = \{e_i | 1 \leq i \leq n\}$ is the set of all activity identifiers in an event trace $\langle e_1, \dots, e_n \rangle$,
- $L \subseteq I$ is a log file, i.e. a set of event traces, and
- $\Gamma = \mathcal{P}(L)$ is the set of all such log files.

For generating reference models, it is very important to gather log files from various systems executing the process in question. The selection of sites depends on the purpose of the model that should be created. If a model should represent configuration options of a software that is distributed internationally, various sites running successful implementations of the software should be chosen from different countries. If a model should represent good examples for a certain process, various successful implementations of that process should be chosen, but these do not necessarily need to be implemented using the same software. All in all, *the source of the used log files should widely cover the targeted scope and all aspects of the model which should be created.* Let us thus in the following assume that a comprehensive set $L^{raw} = \{L_i^{raw} | 1 \leq i \leq n\}$ of n such raw input log files is available (see the far-left of Figure 1).

Although log files are widely available today, the purpose of their creation and their level of details varies. While the transaction management of databases

requires very explicit and detailed logs for being able to undo all changes completely automatically, sometimes log entries are rather unspecific debug messages introduced and never removed by the programmer to find errors in the code. In any way, the log files are rarely created for deriving process models.

For that reason, the available log files must be pre-processed before we can use actual mining techniques to discover meaningful behavioral patterns, i.e. the log files have to be cleaned of irrelevant data and relevant data has to be aligned. When building configurable reference process models, three aspects are especially important in this phase.

- At first, the data in the log files has to be anonymized. Log files usually contain a lot of personal data. This information is highly confidential and the usage is in most cultures strongly restricted through privacy rights and laws. As configurable reference models target at the re-use by others, it is especially important that no personal information is retained in the model. Hence, the elimination of such personal information should take place before any data is processed.
- Secondly, the level of details of the log files has to be balanced among the different input log files and adjusted to the level targeted for the resulting model by aggregating related log events. Otherwise, the level of details in the generated process model will later on be highly inconsistent among different process branches. To reach this balanced level of details an ontology can for example be used. Then single or groups of log events can be mapped onto an agreed level of ontology classes.
- As the same ontological concept is hardly called in the same way by different sources, it must also be ensured that log events from the different source log files are mapped onto each other. This might already come with the use of a common ontology for adjusting the level of details. Otherwise, a direct matching of event names might also be possible.

Further details on how to perform such pre-processing steps can, e.g., be found in [4,12,17]. Process and data mining efforts — as also described in the remainder of this paper — heavily depend on the quality of the pre-processed log files. Therefore, pre-processing comprises in general 60–80 percent of the whole processing efforts [4,17].

In the following we say that pre-processing is a function $prep : \Gamma \rightarrow \Gamma$ which performs all mentioned pre-processing steps for each log file, including a re-naming of log events belonging to the same ontology class to a common class name. The result of the pre-processing is then a consistent set of log files $L = prep(L^{raw})$.

3 Process Mining

The pre-processed log files can serve as the input for a process mining algorithm. Process mining has proven to be a valuable approach for gaining objective insights into business processes which are already in place in organizations. Such

algorithms search for re-occurring patterns in the execution traces of the system in question and generalize the overall process behavior as process models.

To depict process models we will use a workflow net representation here. While the process model types used by existing process mining algorithms vary, most of these models can be transformed into workflow nets using appropriate transformation algorithms as, e.g., provided by the process mining framework ProM [17]. Workflow nets are a formal notation to represent business processes based on Petri nets. Within workflow nets, transitions depict the activities that should happen during the process execution. Through arcs transitions can be connected to places which then represent the conditions resulting from the execution of the transitions. Places also represent the pre-conditions for the execution of transitions whenever an arc connects a place to a transition. A special input place always represents the start of the process, while a special output place represents its completion.

Definition 2 (Workflow net). *A workflow net is a triple $M = (P, T, F)$, such that:*

- P is a set of places,
- T is a set of transitions ($P \cap T = \emptyset$),
- $\mathbf{i} \in P$ is the unique input place,
- $\mathbf{o} \in P$ is the unique output place,
- $F \subseteq (P \setminus \{\mathbf{o}\} \times T) \cup (T \times P \setminus \{\mathbf{i}\})$ is a set of arcs (flow relation), and
- Δ is the set of all workflow nets

Process mining helps process analysts to determine the processes executed by organizations either to document or to improve them. Although reference models are derived from well-running systems, this does not imply that these processes are documented by models, correctly describing the executed behavior. Thus, process mining can also help the reference model designer who has log files from successful process implementations available to create process models.

For this paper we define a mining algorithm as follows:

Definition 3 (Mining algorithm). *A mining algorithm α maps a log file onto a workflow net, i.e.*

$$\alpha : \Gamma \rightarrow \Delta.$$

We thus abstract from the particularities of the mining algorithm, i.e. α may be any process mining algorithm [2]. Further on, we assume that the result of the algorithm fulfills the requirements of a workflow net. This is trivial to achieve for any algorithm that provides a Petri net (or a model that can be transformed into a Petri net) by connecting a unique input place to all its initial elements and a unique output place from all its final elements [5].

For each log file $L_i \in L$ used for the creation of a reference model, a process mining algorithm can therefore generate a process model $M_i = \alpha(L_i)$ depicting the behavior of the log file's original system (see top of Figure 1). Simplified [1],

¹ The description here provides a brief idea of what a process mining algorithm does. In practice process mining is far more complex as the algorithms, e.g., have to take concurrency, incomplete logs, noise, or invisible tasks into consideration [2].

a process mining algorithm splits a log file into the event traces of individual cases, i.e. process instances. It then constructs the process model by analyzing and comparing the events in the traces. Each log event is mapped onto a corresponding transition in the model. For each event that occurs in the event trace, the algorithm checks in the so-far derived model if the corresponding transition can be reached from the transition corresponding to the preceding log event. If this is not the case, a choice is introduced at the place succeeding the transition corresponding to the preceding log event by adding a new arc leading from this place to the transition corresponding to the event in question. The resulting process model will thus depict that when reaching the particular point of the process, the process flow can either continue as all the previous traces did or it can continue as this deviating event trace did.

After having derived a process model M_i for each log file $L_i \in L$, the process model designer can compare these models with each other. By manually or automatically aligning and merging them, an integrated process model $M_{1..n}^+ = M_1 \oplus M_2 \oplus \dots \oplus M_n$ representing the behavior of all the individual models can be generated [9].

However, instead of deriving an individual process model for each system and later on trying to integrate these models, process mining algorithms can also be used to directly generate an integrated model for all of the log files in L . If we concatenate all the log files $L_i \in L$ into a single log file $L_{1..n} = \bigcup_{i=1..n} L_i$, the process mining algorithm α still works in exactly the same way on $L_{1..n}$ as it did for each of the individual log files. Due to the alignment of event names in the pre-processing, the algorithm is able to recognize which log events belong to the same class of events and match them. Thus, the algorithm just processes more process instances and creates a process model $M_{1..n} = \alpha(L_{1..n})$ that is valid for all these instances. That means, the resulting model usually contains more choices than each of the individual models M_i because a combined set of event traces might contain more variants than a subset of these traces. But, as this model represents the behavior of all the instances from the various systems, the model is in the same way an integrated process model valid for all the different input systems as a model generated from merging the individual models [2].

Two properties are important when selecting a process mining algorithm for building such a process model integrating various process variants.

Among different systems it is well possible that steps executed in the process of one system are skipped in the other system. In such cases, the process mining algorithm must be able to introduce a by-pass for the skipped step in the generated process model, e.g. through adding a so-called invisible or silent transition as an alternative to the skipped transition. The invisible transitions then allow for state changes without corresponding to any log events and thus without representing any ‘real’ behavior.

² While the model created by merging several individually mined models should in theory represent the same behavior as the integrated model mined from the combined set of log files, the resulting models depend on the used process mining and merging algorithms and will thus hardly be identical in practice.

Further on, it might later on be desired that a process model can be derived from the reference model which does not conform exactly to the behavior of one of the systems used for the development of the reference model. Instead it might be necessary to combine various aspects of different systems which requires that the used process mining algorithm over-approximates the behavior of the input systems. Most process mining algorithms achieve this as they analyze choices between events only locally and neglect dependencies between choices that do not directly follow each other. By neglecting such non-local non-free choices, the resulting process models permit for example to chose in the beginning of the process a process part that only occurred in a subset of the process instances, while at a later stage a choice is made for a process part that was not part of any of these process instances.

An overview of process mining algorithms is provided in [2] while the ProM process mining framework [117] provides implementations for many of these algorithms. The choice for a concrete algorithm and the quality of the resulting model very much depends on the input log files [15]. In controlled experiences with high-quality input data, we achieved good results using the multi-phase miner [5] because it guarantees the fitness of all the event traces to the resulting model. Using real-world data, it is however hardly possible to derive such high-quality log files during the pre-processing. In such cases algorithms that are able to deal with “noise” in the log files might perform better.

4 Deriving Configurations

The mined process model allows for the execution of all the process’s various variants as it is based on the execution log files from the varying systems. Compared to a set of different models, the integrated model has the advantage for the process designer that later maintenance changes only need to be performed once on the integrated model, and not various times for each of the process variants. The integrated model also covers all combination possibilities of process parts which is usually impossible to achieve when providing a set of process variants. Naturally, reference model users do not need all these variants. Instead, they like to have a specific model covering the individually needed process behavior. Hence, the reference model user needs to configure the integrated model to that subset which depicts this desired behavior.

To define such a configuration for workflow nets, we simply say that a configuration is the set of all elements that should remain in the workflow net. In this way, the configured net can be derived by creating the intersections of the workflow net’s elements with the configuration.

Definition 4 (Configuration). *Let $M = (P, T, F)$ be a workflow net. Then any $C \subseteq P \cup T \cup F$ such that $\{\mathbf{i}, \mathbf{o}\} \subseteq C$ and $\forall (n_1, n_2) \in F \cap C \{n_1, n_2\} \subseteq C$ is a configuration of M . $M_C = (P \cap C, T \cap C, F \cap C)$ is the configured workflow net using configuration C .*

Of course, a careful selection must be made for the configuration as many configurations are not feasible, e.g. because they would eliminate the execution of tran-

sitions that are essential for the process. For example, it is obviously impossible to check an invoice during an invoicing process, if the invoice has not been created beforehand. That means that in addition to the integrated model, the reference model provider also needs to offer some guidance to ‘good’ configurations.

Examples for such good configurations are the systems used to create the integrated process model. These established variants of the process could thus provide a better starting point for reference model users that want to derive their individual models than the complete integrated model of all process variants can be. If we know the configurations of the integrated model leading to the selected, established process variants, and if we know which of these variants might probably be the closest to our requirements (e.g. because of a comparable company size and target market) then the derivation of an individual process would normally just mean to slightly amend this given configuration by adding a number of elements from the integrated model to the configuration and/or removing some of them. In this way, the risky and time-consuming task of configuring the process from scratch can be avoided.

To determine such a configuration, we can re-use the (cleaned) log file of the particular system. It contains all behavior possible in the particular system and can be ‘re-played’ on the integrated model. To depict how this re-play is performed, we first need to introduce the concept of a path of a workflow model.

Definition 5 (Path). *Let $M = (P, T, F)$ be a workflow model. Then $\Phi = \{\langle n_1, \dots, n_m \rangle \in (P \cup T)^* \mid (\forall_{i=1..m-1} (n_i, n_{i+1}) \in F)\}$ is the set of paths of M . The set of elements of a path $\langle n_1, \dots, n_m \rangle \in \Phi$ is defined by the function $elements : \Phi \rightarrow \mathbb{P}(P \cup T \cup F)$ such that $elements(\langle n_1, \dots, n_m \rangle) = \{n_1, (n_1, n_2), n_2, (n_2, n_3), \dots, (n_{m-1}, n_m), n_m\}$.*

To depict the re-play we assume that the integrated model was created by a mining algorithm like the simplified algorithm depicted in Section 3 which guarantees a fitness of 1, i.e. that the behavior of all traces of the log file L_i are represented by the integrated model, and that aspects like concurrency, noise, or incompleteness are neglected. In this way, the re-play starts for each trace of the log file from the input place \mathbf{i} of the integrated model and searches for a path to a transition that corresponds to the first log event of the particular trace. This path should however not pass any visible transitions as their occurrence would require a corresponding event in the log file before the first event. Next, a path through places and invisible transitions is searched from this transition onwards to the next log event and so on. When the transition corresponding to the last log event of an event trace is found, the replay must conclude with finding a path from this last transition to the output place \mathbf{o} . This process is repeated for every trace in the log file. The configuration of the model corresponding to the behavior of all these traces is then the set of all the transitions, places, and arcs used during the re-play. The individual model corresponding to this behavior can then be derived from the integrated model as depicted in Definition 4 and all unnecessary elements can automatically be dismissed.

Definition 6 (Log replay). Let $M_{1..n} = (P, T_{vis} \cup T_{inv}, F)$ be a workflow net with T_{inv} as its set of invisible transitions, and let L_i be a log file. Moreover, let $\bigcup_{\theta \in L_i} events(\theta) \subseteq T_{vis}$ and $\Phi' = \{(n_1, \dots, n_m) \in \Phi \mid \{n_1, n_m\} \in T_{vis} \cup \{\mathbf{i}, \mathbf{o}\} \wedge \{n_2, \dots, n_{m-1}\} \subseteq T_{inv} \cup P \setminus \{\mathbf{i}, \mathbf{o}\}\}$. Then

$$C_i = \bigcup \{elements(\langle \mathbf{i}, \dots, e_1 \rangle) \mid \langle \mathbf{i}, \dots, e_1 \rangle \in \Phi' \wedge \langle e_1, \dots \rangle \in L_i\} \\ \cup \bigcup \{elements(\langle e_k, \dots, e_{k+1} \rangle) \mid \langle e_k, \dots, e_{k+1} \rangle \in \Phi' \wedge \langle \dots, e_k, e_{k+1}, \dots \rangle \in L_i\} \\ \cup \bigcup \{elements(\langle e_m, \dots, \mathbf{o} \rangle) \mid \langle e_m, \dots, \mathbf{o} \rangle \in \Phi' \wedge \langle \dots, e_m \rangle \in L_i\}$$

is the configuration of $M_{1..n}$ that corresponds to the behavior represented by L_i .

While the re-play of log files on models that were created using more sophisticated process mining algorithms is more complex (e.g. a log event might not have a corresponding transition or the path to the next log event might start from a different transition which corresponds to an earlier log event) [14], the configuration can still be discovered by simply identifying the visited model elements. An algorithm to perform such a complex re-play is for example part of the conformance checker provided by ProM.

5 Conclusions

In this paper we showed how reference process models can be constructed from log files of established business processes. Derived by proven process mining algorithms, these models depict and integrate the behavior of several different variants of a common business process. Such an integrated model can afterwards be restricted to the individual process variant required by an organization by means of configurations. By re-playing a log file of an existing system on the integrated model, a configuration of the model conforming to the system's behavior can be identified. Such configurations can serve as starting points for individual configurations. While the development of process mining and conformance checking methodologies mainly aimed at understanding and improving existing systems, they also proved to be very useful for aligning various, successfully running systems during our experiments. By highlighting the individual configurations on the model, users can detect similarities and differences among the process variants as well as new possible configurations far easier than if they have to compare separate models.

In future research we have to setup experiments with larger real-world data to provide further guidance into the practical usage of the suggested methods as well as to test the applicability of further machine learning techniques. For example, we expect that the mining of association rules among different configurations can provide insights on interdependencies between configuration decisions and thus be used for further guidance on making configuration decisions. The re-play of log files generated from systems that are already based on a configurable reference model might help improving the configurable model over time.

Acknowledgements. We thank Antal van den Bosch and Ton Weijters for providing us with insights into the various machine learning techniques.

References

1. van der Aalst, W.M.P., van Dongen, B.F., Günther, C.W., Mans, R.S., Alves de Medeiros, A.K., Rozinat, A., Rubin, V., Song, M., Verbeek, H.M.W., Weijters, A.J.M.M.: ProM 4.0: Comprehensive Support for Real Process Analysis. In: Kleijn, J., Yakovlev, A. (eds.) ICATPN 2007. LNCS, vol. 4546, pp. 484–494. Springer, Heidelberg (2007)
2. Alves de Medeiros, A.K.: Genetic Process Mining. PhD thesis, Technische Universiteit Eindhoven (2006)
3. Becker, J., Delfmann, P., Knackstedt, R.: Adaptive Reference Modelling: Integrating Configurative and Generic Adaptation Techniques for Information Models. In: Reference Modeling, pp. 27–58. Springer, Heidelberg (2007)
4. Cabena, P., Hasjinian, P., Stadler, R., Verhees, J., Zanasi, A.: Discovering data mining: from concept to implementation. Prentice-Hall, Upper Saddle River (1998)
5. van Dongen, B.F., van der Aalst, W.M.P.: Multi-Phase Mining: Aggregating Instance Graphs into EPCs and Petri Nets. In: Proceedings of the Second International Workshop on Applications of Petri Nets to Coordination, Workflow and Business Process Management, pp. 35–58. Florida International University, Miami, FL, USA (2005)
6. van Dongen, B.F., Jansen-Vullers, M.H., Verbeek, H.M.W., van der Aalst, W.M.P.: Verification of the SAP Reference Models Using EPC Reduction, State-space Analysis, and Invariants. *Computers in Industry* 58(6), 578–601 (2007)
7. Eindhoven University of Technology. The ProM framework, <http://prom.sf.net/>
8. Fettke, P., Loos, P.: Classification of Reference Models – a Methodology and its Application. *Information Systems and e-Business Management* 1(1), 35–53 (2003)
9. Gottschalk, F., van der Aalst, W.M.P., Jansen-Vullers, M.H.: Merging Event-driven Process Chains. BPM Center Report BPM-08-08, BPMcenter.org (2008)
10. Gottschalk, F., van der Aalst, W.M.P., Jansen-Vullers, M.H., La Rosa, M.: Configurable Workflow Models. *International Journal of Cooperative Information Systems* 17(2), 177–221 (2008)
11. Keller, G., Teufel, T.: SAP R/3 Process-oriented Implementation: Iterative Process Prototyping. Addison Wesley Longman, Harlow (1998)
12. Pyle, D.: Data Preparation for Data Mining. Morgan Kaufmann, San Francisco (1999)
13. Rosemann, M., van der Aalst, W.M.P.: A Configurable Reference Modelling Language. *Information Systems* 32(1), 1–23 (2007)
14. Rozinat, A., van der Aalst, W.M.P.: Conformance Checking of Processes based on Monitoring Real Behavior. *Information Systems* 33(1), 64–95 (2008)
15. Rozinat, A., Alves de Medeiros, A.K., Günther, C.W., Weijters, A.J.M.M., van der Aalst, W.M.P.: The Need for a Process Mining Evaluation Framework in Research and Practice. In: ter Hofstede, A.H.M., Benatallah, B., Paik, H.-Y. (eds.) BPM Workshops 2007. LNCS, vol. 4928, pp. 84–89. Springer, Heidelberg (2008)
16. Thomas, O., Hermes, B., Loos, P.: Towards a Reference Process Model for Event Management. In: ter Hofstede, A.H.M., Benatallah, B., Paik, H.-Y. (eds.) BPM Workshops 2007. LNCS, vol. 4928, pp. 443–454. Springer, Heidelberg (2008)
17. Zhang, S., Zhang, C., Yang, Q.: Data Preparation for Data Mining. *Applied Artificial Intelligence* 17(5), 375–381 (2003)

Interoperability Maturity Models – Survey and Comparison –

Wided Guédria^{1,2}, Yannick Naudet¹, and David Chen²

¹ CITI, Henri Tudor Public Research Center, 29, Av. John F.Kennedy,
1855 Luxembourg-Kirchberg, Luxembourg.

² IMS-LAPS/GRAI, University Bordeaux 1, 351, cours de la libération,
33405, Talence cedex, France
{wided.guedria,yannick.naudet}@tudor.lu
{wided.guedria,david.chen}@laps.ims-bordeaux.fr

Abstract. Developing interoperability implies defining metrics to measure the degree of interoperability between systems. One of the measures is concerned with the use of maturity models, describing the stages through which systems should evolve to reach higher completeness in the realization of a given objective. This paper reviews the main maturity models that are or could be used for interoperability measure, comparing their different aspects in order to evaluate their relevance and coverage with respect to enterprise interoperability.

Keywords: Interoperability measure, maturity models, assessment, enterprise interoperability.

1 Introduction

Nowadays, Information Technology (IT) as well as human systems evolve in a worldwide heterogeneous environment and work in network. For enterprises, operating in such environment requires flexibility and co-operations between other enterprises, sharing their core competencies in order to exploit the market opportunities. Exchanges are needed for both operational control, and to a larger extend for the decision making process during the establishment of cooperation, including opportunity exploration, co-operation planning and implementation.

It is now admitted that a major issue in global collaboration and co-operation is the development of interoperability between enterprise systems. But what is the meaning of interoperability? Numerous definitions of interoperability have been proposed in the literature [1], [2]. In our research work [10], we consider a general systemic approach to interoperability, where interoperability is first viewed as a problem to solve: *An interoperability problem appears when two or more incompatible systems are put in relation* [3]. Then, when taking the view of interoperability as a goal to reach, we can also write: *Interoperable systems operate together in a coherent manner, removing or avoiding the apparition of related problems.*

Now more than ever, interoperability is becoming a key factor to success in organizations or enterprises. Interoperability between systems thus requires considerable attention to be assessed and continuously improved. There exist maturity models, standards, methodologies, and guidelines that can help an organization, an enterprise, or more generally a system, improving the way it operates with others and thus achieving desired interoperability objectives.

Many existing maturity models have been developed for different purposes. Few of them are relevant to interoperability assessment. Some comparison study has been reported in [4]. In this paper, we propose to evaluate main interoperability maturity models with respect to the Framework for Enterprise Interoperability (FEI) currently in the process of standardisation (CEN/ISO 11354), comparing their different aspects, in order to evaluate their coverage of the interoperability domain and focus.

The paper is structured as follows. In section 2, we survey the main maturity models, starting from the well known ISO/15504 (SPICE) [5] considered as a reference in maturity models, and continuing with the models dedicated to interoperability: LISI (Levels of Information System Interoperability) [6], OIM (Organizational Interoperability Model) [7], LCIM (Levels of Conceptual Interoperability Model) [8], and EIMM (Enterprise Interoperability Maturity Model) [9]. In section 3, we tentatively compare the presented maturity models and highlight, for each model, its relevance and coverage to the interoperability domain. Finally we conclude in section 4, and propose future work.

2 Survey of Main Maturity Models

In this section, we present the main maturity models relevant to interoperability domain. SPICE (ISO/IEC 15504) is a general model for processes maturity, not designed specifically for interoperability. However a higher maturity of enterprise processes also contributes to develop process interoperability between enterprises. Consequently SPICE is also included in this survey. Another well known model “CMM” is one of the inputs to ISO/IEC 15504 and is not mentioned separately in this survey.

2.1 SPICE/ISO15504

ISO/IEC 15504 also known as SPICE (Software Process Improvement and Capability dEtermination) is an international standard for software process assessment, developed by the Joint Technical Subcommittee between ISO (International Organization for Standardization) and the IEC (International Electrotechnical Commission). SPICE defines the requirements and basis to define an assessment model for processes. Also often cited is the CMMI model [11], which can be considered as an instance of SPICE and thus will not be discussed here. The six levels of capability defined in SPICE are shown in table 1.

Table 1. ISO/IEC 15504 Capability levels

Level	Name	Description
5	Optimizing process	The implemented Process becomes itself subject of innovation and continuous improvement.
4	Predictable Process	The defined process is performed consistently in practise with a quantitatively known quality.
3	Established Process	The process is performed and managed using a defined process in an organization-wide standard process.
2	Managed Process	The process is planned and tracked with an acceptable quality within defined timescales.
1	Performed Process	The purpose of the process is generally achieved. The achievement may not rigorously planned and tracked.
0	Incomplete Process	There is general failure to attain the purpose of the process. There are no easily identifiable outputs of the process.

2.2 LISI: Levels of Information System Interoperability

The LISI maturity model was developed as one of the C4ISR universal reference resources to define interoperability between information systems. It focuses on technical interoperability and the complexity of interoperations between information systems. LISI considers five levels, describing both a level of interoperability and the environment in which it occurs, as shown in table 2.

Table 2. LISI maturity levels

Interoperability Level	Environment	Description
Enterprise	Universal	Data and applications are fully shared and distributed. Data have a common interpretation regardless of format.
Domain	Integrated	Information is exchanged between independent applications using shared domain-based data models.
Functional	Distributed	Logical data models are shared across systems
Connected	Peer-to-peer	Simple electronic exchange of data.
Isolated	Manual	Manual data integration from multiple systems.

2.3 OIM: Organizational Interoperability Maturity Model

The Organizational Interoperability Maturity Model deals with the ability of organisations to interoperate. It extends the LISI model to assess organizational issues at business/company interoperability level. Five levels are identified, as shown in table 3.

OIM aims at assessing the systemic interoperability of an organisation, considering the quality of its components inter-operations. Only the organizational interoperability is concerned. However, as organizations often require technical interoperability [7], a complementary model (e.g. LISI) will be used.

OIM does not explicitly propose a specific approach to solve interoperability problems but within each level there are some guidelines mainly focusing on the use of

Table 3. The OIM maturity levels

Level	Name	Description
4	Unified	The organization is interoperating on continuing basis. Command structure and knowledge basis are shared.
3	Integrated	Shared value systems and goals, a common understanding to interoperate. However there are still residual attachments to a home organization.
2	Collaborative	Recognised interoperability frameworks are in place. Shared goals are recognised. Roles and responsibilities are allocated but the organizations are still distinct.
1	Ad hoc	Some guidelines to describe how interoperability will occur but essentially the specific arrangements are still unplanned. Organisations remain entirely distinct.
0	Independent	Organisations work without any interaction. Arrangements are unplanned and unanticipated. No formal frameworks in place. Organizations are able to communicate e.g. via phone, fax and face-to-face meetings.

common terms and structures. We can thus assume that OIM proposes integrated or unified approaches.

In an enterprise context, OIM covers business area of concern: with its strong focus on organisational issues, it does not address technical, semantic or syntactical aspects.

2.4 LCIM: Levels of Conceptual Interoperability Model

In [8] the Levels of Conceptual Interoperability Model (LCIM) is proposed to address levels of conceptual interoperability that go beyond technical models like LISI. The layers of the LCIM model are presented in table 4.

Table 4. LCIM maturity levels

Level	Name	Description
4	Harmonized data	Semantic connections are made apparent via a documented conceptual model underlying components.
3	Aligned dynamic data	Use of data is defined using software engineering methods like UML.
2	Aligned static data	Common reference model with the meaning of data unambiguously described.
1	Documented data	Shared protocols between systems with data accessible via interfaces.
0	System specific data	Black boxes components with no interoperability or shared data.

2.5 EIMM: Enterprise Interoperability Maturity Model

The EIMM maturity model was elaborated by ATHENA (Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications)

Table 5. EIMM maturity levels

Level	Name	Description
4	Optimizing	Enterprise systems are systematically traced to enterprise models and innovative technologies are continuously researched and applied to improve interoperability.
3	Interoperable	Enterprise models support dynamic interoperability and adaptation to changes and evolution of external entities.
2	Integrated	The enterprise modelling process has been formally documented, communicated and is consistently in use.
1	Modelled	Enterprise modelling and collaboration is done in a similar way each time, the technique has been found applicable. Defined meta-models and approaches are applied, responsibilities are defined.
0	Performed	Enterprise modelling and collaboration is done, but in an ad-hoc and chaotic manner.

Integrated Project [9], as a maturity model to perform assessments for interoperability in the enterprise domain. EIMM defines five maturity levels as shown in table 5.

3 A Tentative Comparison of Maturity Models

This section develops a tentative comparison of the maturity models presented previously. The Framework for Enterprise Interoperability (FEI) initially developed under INTEROP NoE and now considered as a coming ISO standard (CEN/ISO 11354) is used as the reference to assess the maturity models. This framework also covers the interoperability concepts defined in EIF (European Interoperability Framework) [12] in eGovernment area. The comparison criteria are based on the three main dimensions of the framework (FEI): (1) Interoperability barriers identifying main incompatibility problems between enterprise systems; (2) Interoperability concerns defining areas in which interoperability takes place; (3) Interoperability approaches defining the ways in which solutions can be developed to remove interoperability barriers.

The evaluation uses the following notations. The ‘+++’ means there is a strong concern and the model meets better the criteria, ‘+’ denotes that is weak and ‘++’ is in between, ‘-’ means that the model does not meet or address the criteria.

To better understand the above maturity models, it is interesting to compare first the background of developers behind the models. Four main relevant scientific areas that contribute to interoperability engineering are: computer science, production engineering, management & organization, system theories. Table 6 summaries such a comparison.

Table 6. Scientific background behind the models

	SPICE	LISI	OIM	LCIM	EIMM
Computer science	+	+++	-	++	+
Production engineering	+	-	-	-	+++
Managt.& organisation	+++	-	+++	-	+++
System theories	-	-	-	-	-

Table 6 shows that system theory is not well taken into account in these maturity models. In a recently published Roadmap for enterprise interoperability by European Commission, science of system theory is considered as one of the existing sciences which should significantly contribute to developing enterprise interoperability.

3.1 Inter Enterprise Interoperability and Interoperability Potential Measures

Based on a systemic view of interoperability [3], we identify two interoperability measures that can be assessed using maturity models: interoperability between two enterprise systems (inter) and interoperability potential with respect to one enterprise.

- The *interoperability between two known systems* is the ability for two particular systems to interoperate. It concerns either a system at the time it is built (how composing systems can be interconnected to reach a given interoperability level?) or an existing system (what is the quality of interoperation between its components?).
- The *interoperability potential* characterises the openness of a system to the external world: it is the ability of a system to inter-operate with unknown systems.

SPICE can contribute to assess process maturity contributing to improve the capability for an enterprise to interoperate with an unknown partner in the future. It also can be used to evaluate collaborative processes between enterprises and contribute to improve interoperability between two existing known systems.

The interoperability potential for a system can be characterised by some properties to be satisfied by the system. Table 10 shows some of these properties and the evaluation of each model. Except the use of standards, existing maturity models need to be further developed to address the interoperability potential measure.

Table 7. Interoperability potential properties

	SPICE	LISI	OIM	LCIM	EIMM
Flexibility to adapt	-	++	-	-	++
Agility to react	-	+	-	-	-
Openness	-	-	-	-	-
Use of standards	+++	+++	+++	+++	+++
Reconfigurability	-	-	-	-	-

According to [6], LISI enables to build a potential interoperability matrix to represent the potential for each system to interoperate with others and displays the level at which the interactions will potentially take place. EIMM is also said to cover potential aspect of interoperability; indeed, in the optimizing level, we find dynamic interoperability and adaptation to changes and evolution of external entities. Other models do not address the potential aspect. Table 11 summarizes the relevance of the considered models with respect to the two types of measures.

Table 8. Interoperability potential vs. inter system interoperability

	SPICE	LISI	OIM	LCIM	EIMM
Interop. Potential	+	++	-	-	+
Interop. inter systems	+	+++	+++	+++	+++

It is to note that the measure of inter systems maturity is concerned with a given interoperability relation between two particular systems. The maturity level reached which characterizes this relation does not apply to the two systems considered separately. In other words, a higher maturity for a given interoperability relation of two particular systems does not mean higher maturity for each of the two systems.

3.2 Interoperability Barriers

According to the Framework for Enterprise Interoperability (FEI), currently under standardization (CEN/ISO 11354) [13], there are three categories of barriers to interoperability:

- *Conceptual* barriers which relate to the syntactic and semantic differences of information to be exchanged.
- *Technological* barriers relating to the incompatibility of information technologies (architecture & platforms, infrastructure...).
- *Organizational* barriers which relate to the definition of responsibility and authority so that interoperability can take place under good conditions.

SPICE tackles these barriers by contributing improving process maturity in general. It does not address explicitly incompatibility problems between systems.

LISI was proposed to deal with the IT technological barriers to interoperability. It was thereafter extended by OIM model to cover organizational problems. At the LISI level 4, data should have a common interpretation regardless of format. Consequently, we can assume some part of the semantic problems is also covered, even if the purpose of LISI remains focused on a technological aspect.

According to [8], the focus of LCIM lies on the data to be interchanged and the interface documentation. The model is intended to be a bridge between conceptual design and technical design: while covering semantic barriers, it also deals with technological one. For EIMM, we can notice that it deals specifically with enterprise assessment, which mainly concerns the organisational barriers to interoperability. It is focusing on the use of enterprise models and the maturity of their usage, which require a correct syntactic and semantic representation [9]. In that way, EIMM is also concerned with semantic problems.

Table 9. Interoperability barriers

	SPICE	LISI	OIM	LCIM	EIMM
Technological	+	+++	-	++	-
Organizational	+	-	+++	-	+++
Conceptual	+	+	-	+++	++

3.3 Interoperability Approaches

Deriving from ISO 14258, we can consider the following three basic ways to relate entities together to establish interoperations [14]:

- The *integrated approach*, characterized by the existence of a common format for all the constituents systems. This format is not necessarily a standard but must be agreed by all parties to elaborate models and build systems.
- The *unified approach*, also characterized by the existence of a common format but at a meta-level. This meta-model provides a mean for semantic equivalence to allow mapping between diverse models and systems.
- The *federated approach*, in which no common format is defined. This approach maintains the identity of interoperating systems; nothing is imposed by one party or another and interoperability is managed in an ad-hoc manner.

Table 9 assesses the maturity models with respect to the above three approaches for interoperability. LCIM proposes to use unified approach: it explicitly proposes solutions to solve interoperability problems, like e.g. the development of a common ontology, common or shared reference models and standardized data elements. Other maturity models do not propose explicitly a solution; they provide a set of basic practices and guidelines to reach each of the maturity levels. The proposed guidelines require conformance and compliance to standards, which can be related to either integrated or unified approaches of interoperability. The use of federated approach to improve interoperability is still missing in existing maturity models and this remains a challenge for future research.

Table 10. Interoperability approaches

	SPICE	LISI	OIM	LCIM	EIMM
Integrated	++	++	++	-	++
Unified	++	++	++	+++	++
Federated	-	-	-	-	-

3.4 Interoperability Concerns

According to [13], there are mainly four areas concerned by interoperability problems in enterprise: data, services, processes and business.

- *Interoperability of data* aims to make work together different data models with different query languages to share information coming from heterogeneous systems.
- *Interoperability of services* aims at making work together various services or applications (designed and implemented independently) by solving the syntactic and semantic differences.
- *Interoperability of processes* aims to make various processes work together. In the interworked enterprise, the aim will be to connect internal processes of two companies to create a common process.
- *Interoperability of business* aims to work in a harmonized way to share and develop business between companies despite the difference of methods, decision-making modes, culture of the enterprises, the commercial making, etc.

SPICE is obviously concerned with the process area. LISI enables information systems to work together and provides assessments for procedures, data, infrastructures

Table 11. Interoperability concerns

	SPICE	LISI	OIM	LCIM	EIMM
Business	+	-	+++	-	+
Process	+++	-	-	-	+++
Service	+	+++	-	-	++
Data	+	+++	-	+++	++

and applications (PAID attributes) within each level. In that way, it covers the interoperability of data and services. LCIM deals with the interoperability of data: it focuses on data model alignment and ‘meaningful’ interoperability. EIMM aims at measuring enterprise model maturity and covers main enterprise model views such as function, service, process, data, information, organisation as well as other aspects such as business strategy, legal environment, security and trust. Table 7 summarises the interoperability concerns dealt by these maturity models.

4 Conclusion and Perspectives

Maturity models aim at helping an organization, enterprise or more generally a system to improve the way it does business, co-operates, or inter-operates with other entities.

In this paper, we surveyed the main interoperability maturity models that exist and presented a tentative comparison of the considered models towards different aspects and criteria of the interoperability problem. This comparison attempt aims to assess the capability of the considered models to measure interoperability potential and their capability to solve a problem of interoperability and to evaluate their coverage of the domain of interoperability. The comparison developed is rather straightforward and further refinement is still necessary.

During this study, we noticed that SPICE could be considered as a generic model, which can be instantiated to the other process oriented maturity models in their specific context. From another point of view, the establishment of interoperability can be considered as a process transforming isolated systems to interoperable systems. In this sense, SPICE can be also used to evaluate the maturity of this process.

The comparison study presented in the paper shows that the existing interoperability maturity models (LISI, OIM, LCIM, EIMM) are partial models only dealing with some aspects of the enterprise interoperability domain as defined in FEI framework. An interoperability maturity model covering all areas of concerns and aspects of the enterprise interoperability is still missing.

There is also a need to identify properties and metrics to allow better characterising and measuring interoperability potentiality. Existing interoperability maturity models were not developed to a satisfactory level to measure explicitly potentiality.

As perspective, SPICE as well as the FEI framework will form the backbone of our future investigation. We will thereafter refer to some of the presented models, in order to establish our model of maturity for system interoperability through the use of a systemic approach [15]. The systemic theory will help us to define a more general maturity model.

References

1. Thomas, C., Ford, J.M., Colombi, S.R., Graham, D.R.: A Survey on Interoperability Measurement. In: 12th ICCRTS, Rhode Island (2007)
2. IEEE Standards Information Network. IEEE 100, The Authoritative Dictionary of IEEE Standards Terms, 7th edn. IEEE, New York (2000)
3. Naudet, Y., Latour, T., Chen, D.: A systemic approach to interoperability formalization. In: IFAC WC 2008, invited session on Semantic-Based Solutions for Enterprise Integration and Networking, Seoul, Korea (July 2008)
4. Panetto, H.: Towards a classification framework for interoperability of enterprise applications. *International journal of Computer Integrated Manufacturing* 20(8), 727–740 (2007)
5. International Organization for Standardization and International Electrotechnical Commission, ISO/IEC 15504 Software Process Improvement and Capability DEtermination Model (SPICE) (2001)
6. C4ISR Interoperability Working Group, Levels of information systems interoperability (lisi), Tech. report, US Department of Defense, Washington, DC (1998)
7. Clark, T., Jones, R.: Organisational interoperability maturity model for c2. In: Proc. of the 1999 Command and Control Research and Technology Symposium, Whashington (1999)
8. Tolk, A., Muguira, J.A.: The levels of conceptual interoperability model. In: 2003 Fall Simulation Interoperability Workshop, USA (September 2003)
9. ATHENA. Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications, FP6-2002-IST1, Integrated Project Proposal (April 2003)
10. Guedria, W.: Decision-support for diagnosis of system's interoperability. In: I-ESA 2008 Doctoral Symposium, 4th International Conference Interoperability for Enterprise Software and Applications, Berlin, Germany (2008)
11. Method Integrated Team.: Standard CMMI Appraisal Method for Process Improvement (SCAMPI), Version 1.1: Method Definition Document Members of the Assessment (2001)
12. COMPTIA. European Interoperability Framework, white paper, ICT Industry Recommendations; Brussels (2004)
13. INTEROP, Enterprise Interoperability -Framework and knowledge corpus- Final report, INTEROP NOE, FP6 -Contact n 508011, Deliverable DI.3 (May 2007)
14. ISO 14258 - Industrial Automation Systems – Concepts and Rules for Enterprise Models, ISO TC184/SC5/WG1 1999-April-14 version (1999).
15. Von Bertalanfy, L.: *General System Theory: Foundations, Development, Applications*. Georges Braziller, Inc., New York (1968)

Using the Zachman Framework to Achieve Enterprise Integration Based-on Business Process Driven Modelling

Javier Espadas, David Romero, David Concha, and Arturo Molina

CIDYT - ITESM Campus Monterrey, Monterrey, Mexico
mijail.espadas@itesm.mx, david.romero.diaz@gmail.com,
a00262912@itesm.mx, armolina@itesm.mx

Abstract. Enterprise interoperability enables the access to relevant information within an enterprise. Traditionally, enterprise integration implementations are managed as merely technological solutions, but integration requirements outcome from business process needs, such as just-in-time information and integration with strategic partners. Without a business process driven methodology that comprises both technical and behavioural views, enterprise integration projects are prone to failure. This paper proposes a business process driven modelling approach for enterprise integration using the Zachman framework to build the enterprise architecture interoperability. The most important model of the enterprise architecture in this proposal is the business process level and its relationships with other architectural model levels.

Keywords: Enterprise Integration, Enterprise Interoperability, Business Process Modelling, Zachman Framework.

1 Introduction

The enterprise integration challenge demands an enterprise capable of exchanging and providing timely and accurate information to both internal (managers) and external (suppliers) stakeholders, and able to integrate its business processes end-to-end across the enterprise and its key strategic partners. Enterprise Integration (EI) means an integration of enterprise business processes through the integration of the information systems that support them to facilitate the effective communication and interaction between their workflows and databases in order to help the enterprise participants in their daily decision-making (Lim et al, 1997). A fully integrated enterprise should be able to rapidly respond to the changing market conditions thanks to the integration of its business functions (business processes) into a single system (workflow) utilizing information technology and sharing data with third-party organisations and customers to externalize and internalize relevant information to support its business operation (Lee et al, 2003).

Enterprise integration benefits for enterprises' business operation include: powerful data mining and enhanced configurable reporting capabilities to support decision-making process (information integration), and increased customer service, heightened market

demands awareness, improved responsiveness to market changes, improved product quality, and better business strategy understanding to enhance the business processes streamlines to achieve organisational efficiency (Shuangxi & Yushun, 2007).

Enterprise integration enhances the enterprise architecture capabilities of any organisation to become a more agile, flexible and robust enterprise in order to achieve both technical and behavioural integration for a greater degree of communication, coordination and cooperation among human actors as well as information systems. Technical integration refers to software and hardware integration, meanwhile, behavioural integration refers to roles, rights and responsibilities redistributions between enterprise participants to support the new organisational structure and information requirements of the stakeholders involved in the enterprises business processes (Lee et al., 2003).

This paper proposes an enterprise integration methodology using Zachman framework to achieve enterprise integration based-on a business process modelling approach that considers technical and behavioural implications thanks to different stakeholders' roles and perspectives provided by the Zachman framework.

2 Basic Concepts

For a complete understanding of the enterprise integration methodology to be proposed, three main concepts should be defined: (1) Enterprise Architecture representing the overall view of the environment in where the enterprise integration (system integration) is going to be performed (the enterprise). (2) Enterprise Integration as the capability to integrate all the components depicted in the enterprise architecture, with special focus on information systems. (3) Business Processes Modelling as a precise description of the integration needs thanks to a modelling exercise that can depict the dependency relationships among people-to-people, system-to-system and people-to-system.

2.1 Enterprise Architecture: Zachman Framework

Zachman defines an enterprise architecture as a set of descriptive elements that are relevant for describing an enterprise in such a way that it can be produced to management requirements and maintained over its useful life period. Besides, an enterprise architecture provides a common view of the enterprise resources (people, processes, technology and other assets) and how they integrate to respond to enterprise drivers (business objectives) (Anaya & Ortiz, 2005). This common enterprise view helps to understand the high-level integration requirement within enterprises.

Another benefit identified from an enterprise architecture is that it enables the discovery and elimination of redundancy in business processes, reducing information systems complexity (Melissa, 1996). In this case it is possible to use business process modelling to identify the enterprise boundaries, the different business value chains, the enabling business processes, the business process map of each value chain and the organisational units participating in them (Anaya & Ortiz, 2005). Previous characteristics help to identify integration needs based-on the enterprise (business) perspective instead of the technological view. Conveniently, it is possible to develop a solution based-on the involved enterprise business processes.

In the Zachman framework, the enterprise architecture is modelled by five roles or perspectives, and they are represented in each row of the framework (Steinke & Nickolette, 2003): (1) Planner, focused on scope; (2) Owner, focused on deliverables; (3) Designer, focused on specifications, (4) Builder, focused on production; and (5) Subcontractor, focused on reusable components. Therefore, the Zachman framework will help to understand the different levels of an integration requirement, from a high level model (business model) to a specific platform (system model) to an enterprise integration solution implementation (technological model).

2.2 Enterprise Interoperability: Information and Applications Interoperability

The term enterprise integration reflects the capability to integrate a variety of system functionalities and resources within enterprise environment. Enterprise integration connects and combines people, processes, systems, and technologies to ensure that the right people and the right processes have the right information and the right resources at the right time (Brosey et al, 2001). Enterprise integration tangible goals include increasing product quality, decreasing unit cost and reducing cycle time while non-tangible goals include timely information access, automatic process monitoring and distributed co-operative working, integrated systems and globalization (Lim et al, 1997). Commonly, enterprise integration solutions are based-on enterprise application integration approaches. Enterprise application integration is a strategic approach for binding many information systems together, at both service and information levels, supporting their ability to exchange information and leverage processes in real time (Jimenez & Espadas, 2007).

Traditional approaches of interoperability integrated peer-to-peer information systems as the business processes needs tend to appear. New solutions have been emerging in order to create better information architectures and interoperability frameworks (e.g. ATHENA <http://www.athena-ip.org> and IDEAS <http://www.ideas-roadmap.net>, EU funded projects). Nowadays, Service Oriented Architecture (SOA) is the common solution implemented inside enterprises. In a typical service-based scenario, a service provider hosts a network-accessible software module (an implementation of a given service) which can be used to accomplish a specific task or activity of a business process (Jimenez & Espadas, 2007).

Enterprise integration is not purely about technology integration. Enterprise integration includes considerations about business processes that spread across various IT applications, providing the overarching basis for technology integration (Lam & Shankaraman, 2007). Enterprise integration is therefore an activity that is business process driven not technology driven, it coordinates business processes with and across different parts of the enterprise and involves multiple stakeholders. Furthermore, it adopts a strategic rather than a tactical or localized view.

2.3 Business Process Modelling: Enterprise Workflows

Agile organisations want to be able to modify their business processes according to the changes in their business climate, including: competitive, market, economical, industrial or other factors. The premise of EAI is that it allows a business-oriented approach to map business processes rather than a technology-driven business reengineering (Lee et al.,

2003). Today’s methodologies of EAI must help to understand the end-to-end business processes that are critical for organisation’s goals. This understanding tell us what processes the organisation needs to integrate and why. It is also important to include business processes that take place between organisations and business partners (Lam & Shankaraman, 2004). This business process approach is the base of the present research because the business processes are conceived as the core model for the enterprise integration architecture.

With a business process approach, it is possible to have a business process manager within an enterprise which sees the company as a set of processes that generate and consume different types of assets that are carried out by resources.

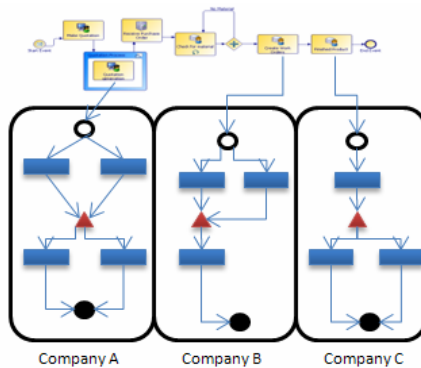


Fig. 1. Business process integration (Lithicum, 2003)

More complex requirements take place when several steps within the business process require information from different applications (see Figure 1), and in some cases applications need to interact and share information with external applications, databases or any other information source (Espadas, 2006). This problem can be defined as a Business Process Integration (BPI) scenario. BPI represents the analysis and methodology of how multiple business process can be integrated and how they can be executed within this integration. Benefits from BPI include (Lam & Shankaraman, 2007): (1) it extends traditional integration with knowledge of business process, (2) integrated with a business process engine allows business processes to be executed, monitored and managed, and (3) captures workflow between IT applications and humans, and information sharing. The relevance of business process approach is that it tries to improve measurable business performance for stakeholders, through ongoing optimization and synchronization of enterprise-wide process capabilities (Business Process Management Group, 2005).

3 Enterprise Integration Methodology

The enterprise integration methodology proposed in this paper uses the Zachman framework to define the enterprise architecture in where the enterprise integration project is going to be performed and a business process modelling approach to

identify the dependency relationships among human actors and information systems that should be integrated and supported to achieve a fully integrated enterprise.

For authors' enterprise integration methodology, the Zachman framework scope and business model rows were the most important ones, since they can offer a list of enterprise business processes and units and their information requirements that should be assessed to identify any lack of information availability within a business process or unit that can trigger a need for information and/or applications integration and interoperability.

The enterprise integration methodology proposed in this paper relies on an *enterprise architecture modelling approach* using the Zachman framework as its first step to identify opportunity areas for enterprise integration and a *business process modelling approach* as its second step to recognize dependency relationships that have to be integrated to become interoperable to support efficient business processes streamlines.

Table 1. Enterprise Architecture based-on Zachman Framework for Business Process Driven Integration

Zachman Row	Model	Artefact	Tool	Description
Scope (Contextual)	Enterprise Business Processes	Business Units	Organisational Charts	Architecting Enterprise & Business Units
Business Model (Conceptual)	Business Process Models	Business Processes	BPMN	Business Process Modelling
System Model (Logical)	Logical Data Models & Application Architecture	Modelling Diagrams	UML & SOA	Service Oriented Design
Technological Model (Physical)	Physical Data Models, Technology Architecture & System Design Imple- mentation	Business Process Execution & Implementation	BPEL & Software Components	Business Process Execution & Integration

Table 1 presents how the Zachman framework (enterprise architecture) is used to define the business process modelling approach to achieve enterprise integration according to each row: (1) the *scope level* or *row* presents a list of entities (business processes and units) that are relevant for the enterprise integration project; (2) the *business model row* refers to a set of business process models (using BPM notation) in where the different business units are represented to show how business process and units are associated according to the business rules; (3) the *system model row* depicts the information and application architectural design requirements to achieve business information systems interoperability, in this case addressed through a software architecture design (SOA) approach; (4) the *technology model row* attends the software and hardware that will be used to achieve the enterprise integration and interoperability requirements, in this occasion authors propose the use of BPEL as the business process execution language for the system design; and (5) the *detailed representation row* addresses the software programs that will be used to code and implement in a specific programming language or platform. Next sections will detail each step/row of the enterprise integration methodology.

3.1 Scope Row: Enterprise Architecting and Business Units Modelling

The first step in the proposed enterprise integration methodology (the scope row) involves the identification and description of the following enterprise elements:

- **Enterprise Business Processes.** Identification of the core-enterprise business processes and their description using a “use case template”.
- **Organisational Units.** Identification of the organisational units, in other words, the main departments within enterprise and their representation in an “organisational chart”.

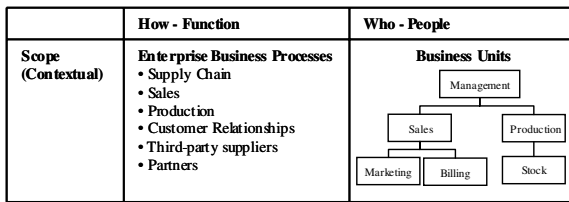


Fig. 2. Zachman Framework Scope Row for Enterprise Integration

Figure 2 depicts a simplified version of *scope row* in where two departments (e.g. sales and production) are involved in the enterprise integration project. These departments can be seen as the business units within the enterprise architecture. The organisational chart can be seen as the high level representation (view) of the integration needs and it can clearly depict the stakeholders and how their activities (responsibilities) are affected by the interoperability issues (requirements).

3.2 Business Model Row: Business Process Modelling

Once that the *scope row* has been modelled, the next step in the enterprise integration methodology proposed is the *business model row* modelling, in which the enterprise business processes and organisational units identified and described in the previous row (*scope model row*) are transformed into a finer granularity model using a business process modelling approach based-on Business Process Modelling Notation (BPMN), but UML 2.0 notation could be also used.

Keeping up with the authors’ previous example, Figure 3 includes the *business model row* results presenting the sales and production business processes models modelled using BPM notation. A pool is used to represent the business units’ boundaries and to allow detecting the business processes integration requirements, when a transition is needed through the different business processes pools. Tasks and decisions are carried out through the business processes in order to accomplish their goals. Also, it is possible to model sub-processes as activities that could be performed within the entire business processes. Besides, BPMN models bring the advantage of automatically transforming the business model into an executable process language know as Business Process Execution Language (BPEL). After the business process modelling activity, a business process analysis is carried out in order to: (1) identify

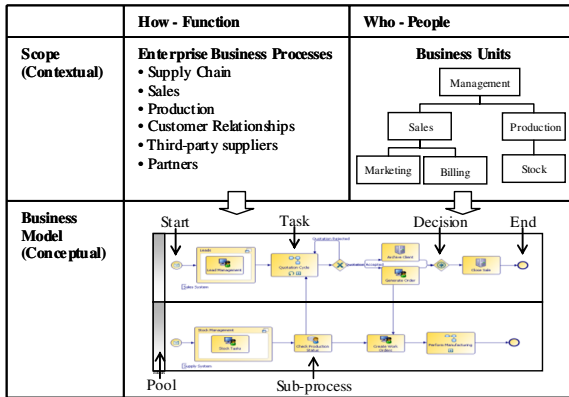


Fig. 3. Zachman Framework Business Model Row for Enterprise Integration

business processes bottlenecks, (2) understand the main activities and categorize their importance, (3) identify business processes integration requirements, (4) analyze information flows through business processes, and (4) make better decisions about business processes models.

Transformation from *scope row* to *business model row* is done by a business process analyst, by decomposing the organisational units and enterprise business processes into business processes models.

3.3 System Model Row: Service Oriented Design (SOA)

Once the *business process model row* has been developed, next modelling level or row is the *system model row*. For this step, the business processes modelled are converted into model artefacts:

- **Logical Data Models.** Information systems representation, including their elements, is performed in this activity. For a complete design and representation of the enterprise information systems, UML (Unified Modelling Language) is recommended as the standard language for system modelling. UML provides a set of diagrams specifications in order to develop the systems' components and behaviours.
- **Application Architecture.** In this activity the information and application architectures are designed based-on a service oriented architecture (SOA) approach. SOA design will express the integration requirements as a set of service providers and clients that have communication. Services are platform-independent components that encapsulate business logic for integration. There are many technologies that can be utilized to implement the service functionalities and platforms for SOA, such as: J2EE, .NET, CORBA, etc. (Shuangxi & Yushun, 2007). Taking into account the business processes modelled in the previous row (*business model row*), the service oriented architecture can be designed based-on the integration requirements of the information systems

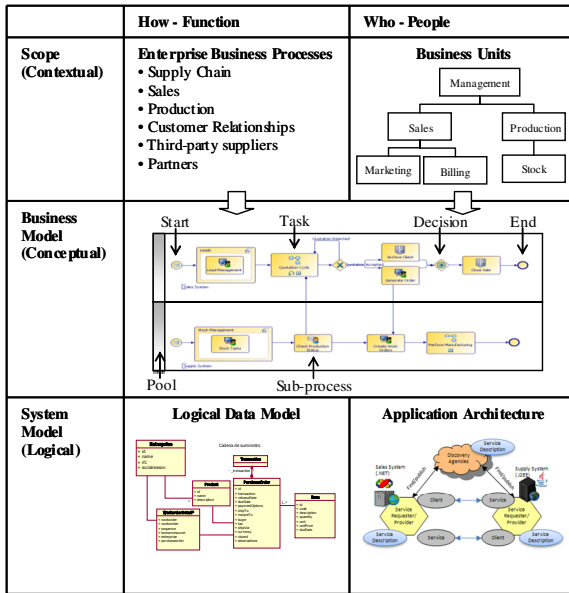


Fig. 4. Zachman Framework System Model Row for Enterprise Integration

that will be supporting the enterprise business processes. SOA implementations rely on XML standards and protocols, such as Web Service Description Language (WSDL) or Simple Object Access Protocol (SOAP).

Continuing with authors’ example, Figure 4 presents the new added row at the framework bottom. The *logical data model* was designed using UML notations such as: class diagrams (for business entities), sequence diagrams (for behaviour), use cases (for functionalities) and/or activity diagrams (for workflow documentation). For the SOA design, each application or software component (e.g. sales application developed in .NET platform) must expose a set of interfaces, conceived as services. Other systems, (e.g. such as supply chain system in J2EE platform), should develop clients or consumers for those services. In some designs, is used a component named “discovery agents”, some kind of yellow pages for all services providers and clients. The providers publish their services interfaces and clients look-up within these agencies for specific services description. Then, the discovery agent returns the service reference to the client, in order to create the communication channel between them.

3.4 Technology Model: Business Process Execution and Integration

Finally, the last level in the authors’ enterprise integration methodology based-on Zachman framework is the *technology model row*. For this row, three activities are carried out to achieve enterprise integration:

- Physical Data Models.** Modelling data for enterprise integration is not a trivial activity. Data integration requires the use of information from different sources, in most cases, these sources are heterogeneous and have different structures (different databases, programming languages or data types). For modelling the data model definition, the use of entity-relationship (E-R) diagrams is proposed to provide a view of enterprise databases and their relations in order to get an holistic perspective of data integration. Once the E-R diagrams are finished, their implementation or integration with specific database managers (DBMS) is developed.
- Technology Architecture.** A business processes execution and integration model is proposed as a specific technical architecture. A wide known execution and integration language is Web Services Business Process Execution Language (BPEL). BPEL defines a model and a grammar for describing the business processes behaviour based-on interactions between the business processes and their partners (other information systems) (OASIS, 2007). The interaction with each partner occurs through Web services interfaces, and the relationship structure at the interface level is encapsulated in what is called a partnerLink. BPEL allows defining multiple service interactions among partners (information systems) in order to coordinate them to achieve a business goal.

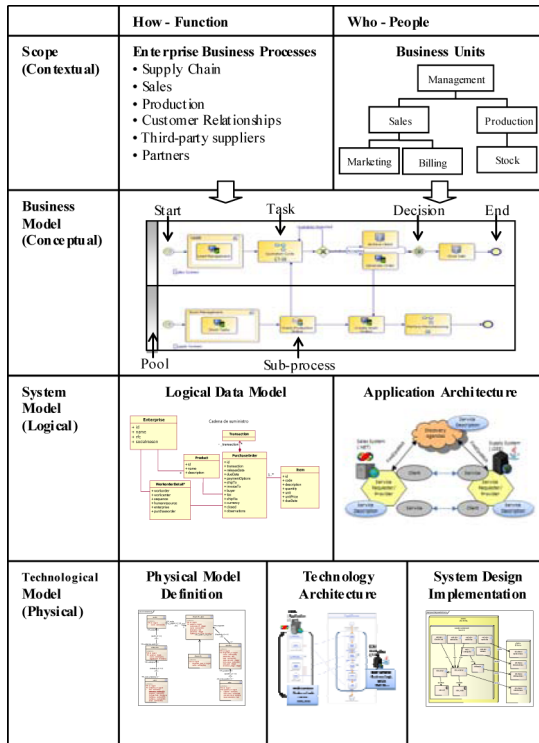


Fig. 5. Zachman Framework Technology Model Row for Enterprise Integration

- **System Design Implementation.** Software components are developed for business processes integration, using the SOA design (at *system model row*) and by building the clients and services interfaces for the specific software platform. The system design implementation is carried out by developing the business processes integration based-on the SOA architecture and the business processes execution language (BPEL). By supposing two heterogeneous platforms (.NET and J2EE in authors' example), the system design implementation activity consists in developing the code to wrap the applications and the clients and services interfaces implementation.

Figure 5 depicts the *technology model row* with the three activities supported in the enterprise architecture. Regarding the system design implementation activity, the business process integration is already achieved and performed and the business processes and applications are now connected and the technology platforms are aware and prepared for interoperability.

4 PyME CREATIVA: A Successful Case Study

Within ITESM University, this methodology was used in PyME CREATIVA project <http://www.pymecreativa.com> - for enterprise integration. This project builds and delivers IT-infrastructure and e-services based-on a transparent, easy-to-use and affordable plug-and-play technological platform, known as the “e-HUB” (Integrated e-Services Center for Virtual Business); under a Software-as-a Service (SaaS) model to Small and Medium Enterprises (SMEs) (for a complete description see: Molina et al., 2006).

Integration requirements were defined based-on SMEs business processes and their current information systems. The integration was developed through SOA design and implemented with Web services technologies. Challenges of this integration project were about the interoperability of different legacy systems in several SMEs with the e-HUB of e-services. These legacy systems were integrated by adapting each one for SOA design, through adapters and wrappers in order to expose their Web services (see more in Molina et al, 2006; Espadas et al, 2007).

5 Conclusions

Enterprise integration projects emerge most of the times from a business need, so it is important to drive the solution as a business requirement. This paper proposes a business process driven approach for facing enterprise integration requirements by using business process models and the Zachman framework as its basis. A set of guidelines and artefacts are presented in the authors' methodology in order to give a path and tools for enterprise integration requirements and changes.

Starting with the highest modelling level of Zachman framework and following it to its lower levels, the business process driven methodology for enterprise integration proposed in this paper provides a complete understanding about the business process importance as the central model in an enterprise integration project.

Acknowledgments

The research presented in this document is a contribution for the Rapid Product realization for Developing Markets Using Emerging Technologies Research Chair, Registration No. CAT077. The authors wish to acknowledge the support of the Innovation Center in Design and Technology from ITESM - Campus Monterrey.

References

1. Anaya, V., Ortiz, A.: How Enterprise Architectures can support Integration. In: Proceedings of the 1st International Workshop on Interoperability of Heterogeneous Information Systems, IHIS 2005 (2005)
2. Brosey, W.D., Neal, R.E., Marks, D.F.: Grand Challenges of Enterprise Integration. In: Proceedings of the 8th IEEE International Conference on Emerging Technologies and Factory Automation, vol. 2, pp. 221–227 (2001)
3. Business Process Management Group: In Search Of BPM Excellence: Straight From The Thought Leader. Towers, S., Fingar, P. (contributors) Meghan-Kiffer Press (2005)
4. Espadas, J., Concha, D., Najera, T., Galeano, N., Romero, D., Molina, A.: Software Architecture Implementation of an e-HUB to offer e-Services for SMEs. *Journal of Research in Computing Science* 31, 193–202 (2007)
5. Lam, W., Shankararaman, V.: An Enterprise Integration Methodology. *IT Professional* 6(2), 40–48 (2004)
6. Lam, W., Shankararaman, V.: Dissolving Organizational and Technological Silos: An Overview of Enterprise Integration Concepts. In: *Enterprise Architecture and Integration* ch. I, pp. 1–22. IGI Global Press (2007)
7. Lee, J., Siau, K., Hong, S.: Enterprise Integration with ERP and EAI. *Communications of the ACM* 46(2), 54–60 (2003)
8. Lim, S.H., Juster, N., de Pennington, A.: The Seven Major Aspects of Enterprise Modeling and Integration: a Position Paper. *ACM SIGGROUP Bulletin* 18(1), 71–75 (1997)
9. Lithicum, D.S.: *Next Generation Application Integration*. Addison-Wesley Professional Press, Reading (2003)
10. Molina, A., Mejía, R., Galeano, N., Nájera, T., Velandía, M.: The HUB as an Enabling Strategy to Achieve Smart Organizations. In: Mezgár, I. (ed.) *Integration of ICT in Smart Organizations*, Hungary, pp. 68–99. IDEA group publishing (2006)
11. OASIS Web Services Business Process Execution Language (WSBPPEL) TC (April 2007), <http://www.oasis-open.org>
12. Shuangxi, H., Yushun, F.: Model Driven and Service Oriented Enterprise Integration - The Method, Framework and Platform. In: *Proceedings of the 6th International Conference on Advanced Language Processing and Web Information Technology (ALPIT 2007)*, pp. 504–509 (2007)
13. Zachman, J.: *Enterprise Architecture: The Issue of the Century*. Database Programming and Design (1997)

Enterprise Modelling Based Services Orchestration

Qing Li, Canqiang Li, and Yun Wang

Department of Automation, Tsinghua University, Beijing 100084, P.R. China
liqing@tsinghua.edu.cn

Abstract. Aiming at the problems of system integration and cross-system interoperability, Service-oriented Architecture (SOA) provides a new integration method and system infrastructure. The key for SOA development and implementation is services encapsulating and orchestrating of applications through certain mechanism so as to operate a complex business. Current workflow technique, business process modelling notation (BPMN) and business process execution language (BPEL) have made very meaningful attempt in this field. But these process modelling technologies have semantics incompleteness when they describe processes in multi-platforms and multi-departments environment. Regarding service-oriented business process modelling technology as the kernel technology, this paper presents the concepts and expression method of process collaboration, proposes services orchestration solution based on process collaboration, specifies service models' expression method and service modelling process through case study, and points out the role of business process modelling in services orchestration.

1 Introduction

As a new enterprise applications integration pattern and infrastructure, SOA(Service Oriented Architecture) encapsulates and releases the business function of the application systems in the form of services so that the interdependent relations between service users and service providers reflects itself only as the service description based on standard document format, completely separating service interfaces from services^[1]. SOA has a couple-loosed nature to meet the requirement of the heterogeneous distributed system integration, which is helpful for enterprise IT resources reuse. SOA provides a flexible solution to business automation, supporting the business agility requirements of enterprises. Therefore, SOA becomes the present focus in the field of the enterprise system integration.

However, to complete a complex business function often requires more than one web service and a mechanism linking multiple independent web services so as to realize an integrated activity with logical relationship. Enterprise modelling technology and languages play a key role in service orchestration.

[3] presents the concepts and principles of SOAD. [4] and [5] introduce a method of implementation of SOA and its main process. [1] makes some amendments to the architecture presented in [5]. However, none of the literatures provide a detailed service modelling method.

[6] presents guidelines for building service model by use-case model based on domain analysis, but the guidelines are too general and need further study. [7] holds that

general business functions summarized according to domain analysis can be used for business service modelling, but it doesn't provide more concrete operation instructions.

[2] gives a complete service modelling process based on service hierarchy, which is very applicable but quite complicated and open to question as a whole.

Enterprise models are used to describe, analyze and design enterprise system and to express some aspects of enterprises. Enterprise models reflect the nature of enterprise operations from different views and form a unified whole [8][9]. And the basic purpose of SOA implementation is to implement enterprise operation automation, thus serving the corporate strategic objectives. Therefore, enterprise model should be an important basis for modelling services.

Mainly on the basis of work referred in [2], a service modelling method based on enterprise models is presented in this paper. First, a definition of service model and service modelling is provided. Second, service model expressing method based on Unified Modelling Language (UML) is discussed^{[10][11]}. And last, based on a project instance, the paper describes service modelling process based on enterprise model in detail.

2 Service Modelling Language

Service model expressing method presented in this section is based on several common graphical models in UML. Detailed texts can be supplements besides graphical language. Several major model elements and model expressions list as follows:

(1) Use UML Class Diagram to express services, as shown in Fig.1. Service names and operations are marked respectively in the Class Diagram. Service names need to be nouns or gerunds, such as "order service", "order information enquiry service". General names for operations are verb phrases, such as "enquire order information".

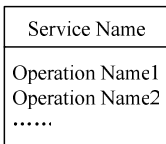


Fig. 1. Single service

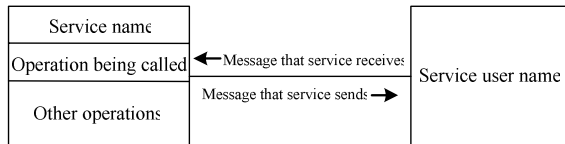


Fig. 2. Interactive relation between services and service users

(2) UML Collaboration Diagram expresses the relationship between services and service users, as shown in Fig.2. Services and service users are connected by solid lines, showing the interactive relationship between them. If the service includes a number of operations, then the operation called by users is isolated from other operations. Receiving and sending messages are noted above and below the lines and the arrows indicate the direction of the transmission of information. As service modeling is at the service analysis stage, there is no need to give very detailed information form, data type and so on, which can be left for detailed designing in the service interface design phase. For example, product inventory enquiry service can provide to

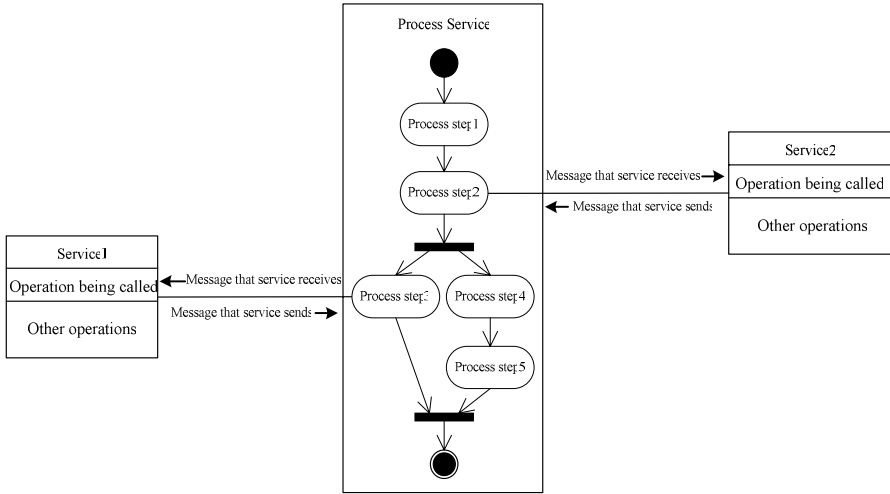


Fig. 3. Interactive relationship between process services and their member services

service users relevant information stored in the information system according to the product name received, and we only have to point out that the information transmitted is product inventory without specifying whether the data is float-type or double-type.

(3) Combining the ideas of UML Activity Diagram and Collaboration Diagram, we express the interactive relationship between process services and their member services, as shown in Fig.3. The relationship diagram of services and service users is the basic form. The process service act as the public service user of all the member services, and it calls the operation in a certain member service in a specific process step, in other words, the operation in a member service achieves a specific process step. To express the relationship clearly, we draw the business process model by using UML Activity Diagram and connect services and corresponding process steps with solid lines, showing that the process steps are achieved by this service operation.

3 Service Modelling Process

The implementation of SOA includes the following phases: services analysis, services design, services realization, services test, services deployment and services maintenance^[2]. Service modelling belongs to the services analysis stage, whose preparatory work includes:

(1) Analyze the enterprise automatic operational application requirements. We mainly consider the requirement of current system integration project, and also consider some relevant or potential requirement to improve the solutions' reusability. Business requirement will be the starting point for service modeling, whose main expression form is business process model.

(2) Understand the information of relevant existing application system or software products to be implemented, primarily their functions and interface methods for

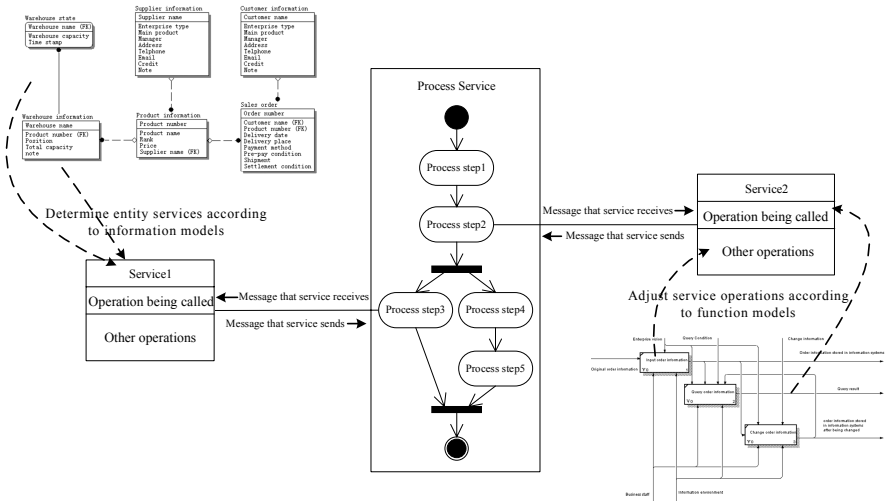


Fig. 4. Demonstration of roles that enterprise models play in service modeling

service modeling reference. Because of no concrete realization involved in the analysis stage, there is no need to understand the technical details.

Enterprise models of different views play different roles in the service modeling process: business process model is the starting point of service modeling, information model is the basis of entity service, and function model is the basis of service operation adjustment, as shown in Fig.4.

Basic modeling process is: taking independent process steps from business process models as alternative service operations; recognizing the operations that should be contained in process service according to business process logic; determining entities related in the business process and compositing corresponding alternative service operations according to enterprise information model; adjusting service operations and finalizing entity services with the reference to entity activities pointed out in enterprise function models and the former results of automatic operational application analysis; considering the technology realization and convenience, project time limitation, cost, etc., to determine task services appropriately; establishing utility services that have nothing to do with specific solutions and are most reusable; adjusting and optimizing the process logic, and determining the interactive relationship between process services and each member service; at last finishing service modeling.

4 Service Modelling Case Study

The authors have carried out the service modelling and development work of a system integration research project based on SOA in a grain distribution enterprise. The following paragraphs specify the service modelling process based on enterprise models with the example of automatic operational application for sale order intention approval (order pre-approval) in this enterprise.

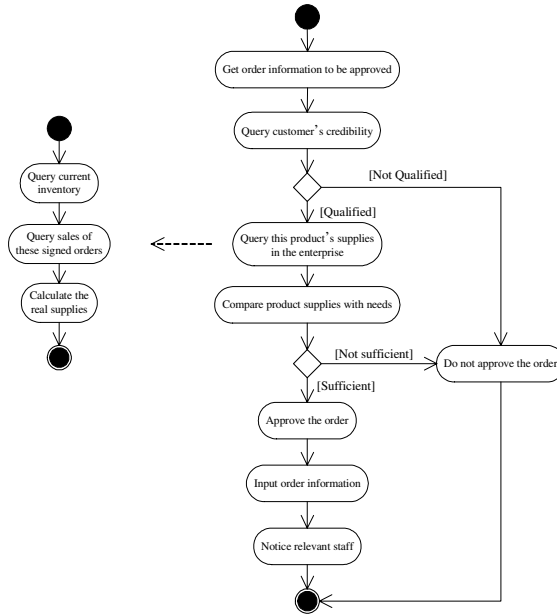


Fig. 5. Order approval business process model of a grain distribution enterprise

As shown in Fig.5., order pre-auditing business process model is described with UML Activity Diagram. After the enterprise salesman reaches an initial agreement with a customer, the enterprise has to check before deciding whether to sign a formal contract mainly based on the customer’s credibility and product supplies. Product supplies can be concluded from current inventory and the sales of those signed orders with an earlier delivery date. If the order is approved, the order information can be input into management information system directly and related business staff can be notified.

As the enterprise hasn’t implemented SOA, there is no web service. Current systems related with process are: records of customer’s credibility included in customer relationship management system (CRM); current inventory information contained in manufacturing field data collecting, real-time storing and comprehensive analysis system of manufacturing execution layer (MES data collecting sub-system); order information contained in order information management system. As these three systems mentioned above don’t provide external interface, information is processed manually through the man-machine interface. However, automated operation can be obtained by developing applications on these systems’ relational database (data layer components).

Steps of service modeling based on enterprise models list as follows:

(1) Get a series of atomic process steps by further subdividing the business process models. The so-called atomic feature mainly refers to an independent unit of work from the perspective of realization, which can’t be further divided.

In this case, “product supply enquiry in the enterprise” is subdivided into three steps: “current inventory enquiry”, “enquiry sales of these signed orders” and “calculation of the possible supplies”, as shown in Fig.5.

(2) Extract alternative service operations from process steps. The following operations should be removed: a) completed manually without the need of automation. For example, the first step “get the order information to be approved”, the information is input by salesman or received from transferred parameters when other applications call the process; b) those already included in other services; c) those not be published as web service.

(3) Recognize operations that should be included in the process services according to the business process logic, to avoid packaging them into business services. Generally speaking, business process logic includes sequence logic, condition logic, exception logic of process steps and some other business rules and necessary process steps.

In this case, business process logic includes that: a) sequence logic of process steps; b) condition logic: if the customer’s credibility is unqualified, end the process and approval fails, otherwise continue; c) process steps: calculate the real supplies from these two results queried; d) process steps: compare real supplies with demanded sales in orders; e) condition logic: if the possible supply is less than the demand, end the process and approval fails. Otherwise approval succeeds, continue. Step c) and d) should not be included by any other business services.

General speaking, what the service users can call are process services functions instead of process services themselves.

(4) Determine entities corresponding to business process according to enterprise information models, and then compose relevant alternative service operations regarding them as logical background.

We build enterprise information models relevant to this process in IDEF1x, and parts of the models are shown in Fig.6. Thus, we can determine three entity services: order, product inventory and customer. The alternative service operations they composed respectively are shown in Fig.7.

(5) Referring to entity activities pointed out in enterprise function models and the former results of automatic operational application analysis, adjust alternative service operations composed in (4) step and establish entity services in order to improve reusability. Concrete adjustment methods: a) change the operation content to make it more universal; b) increase other operations that have nothing to do with this process. If possible, Technology realization method (like functions, calling method and customized development methods of current systems to be encapsulated or software to be implemented) should also be considered to prevent the service models from being too unpractical in practice.

Take order entity as an example; establish order information management of enterprise function models in IDEF0 method, as shown in Fig.8. “Enquiry of order information” can be further subdivided into some common query functions such as “Enquiry of order information on a certain product”, “Enquiry of order information from a certain customer” and so on. According to the requirement analysis, “change order information” doesn’t need to provide function-calling for other application systems, while some common enquiry operations of order information may appear in some other business process. From a view of realization, as current order information

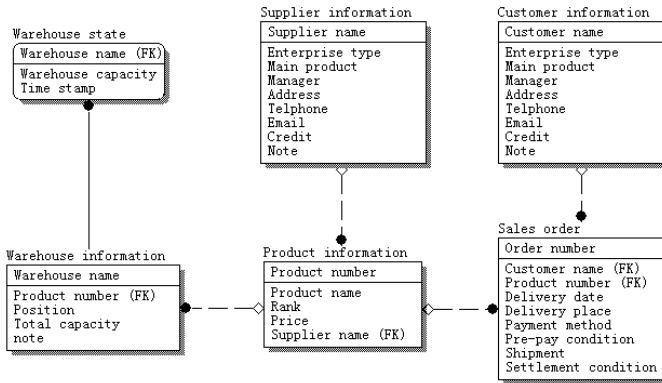


Fig. 6. Parts of the enterprise information models corresponding to the process

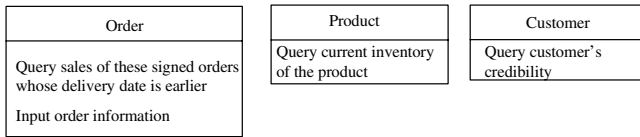


Fig. 7. Alternative service operations entities composite

management systems don't provide function calling interface, we have to develop application programs that can operate their background data directly. Therefore, it is not necessary to worry about whether actual system can match with alternative service operations when they are adjusted.

After considering the above factors comprehensively, operations in order services can be adjusted to: input of order information, some common order information query operations, as shown in Fig.9. The reason changing the original enquiry operations are to make the entity services more reusable. But if we change it as "enquiry of order information", the reusability is improved but the realization ability is reduced, because too large service granularity may affect the actual system performance, which might not be the best choice. For the same reason we can get other two entity services as shown in Fig.9.

(6) Determine task services appropriately to satisfy project requirement, based on the technology realization and convenience, project time limitation, cost and so on.

Further consider the process step: "enquiry of sales of signed orders with an earlier delivery date". The client needs some additional processing if we want to finish this process step by calling operations included in order entity service, which is very cumbersome in program. It is much easier to develop database query program and encapsulate it as service. With limited time, we can only develop this task service and ignore those unrelated operations in entity services. So we decide to add task service of "Enquiry of order sales", in which only the operation "enquiry of the order sales of a certain product in a certain period of time" is included.

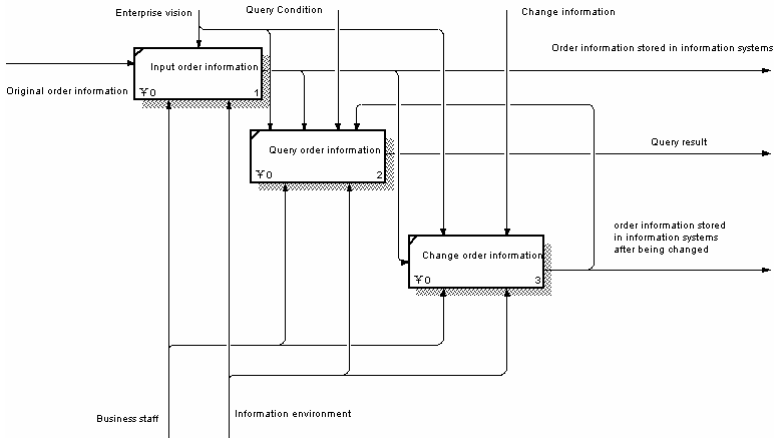


Fig. 8. Order information management in enterprise function models

Order service	Product service	Customer service
Input order information Query order information of a certain product Query sales of these signed orders whose delivery date is earlier Query order information of a certain customer	Query current inventory of the product Query product information	Query customer's credibility Query customer's basic information

Fig. 9. Entity services after adjusting: order service, product service, customer service

(7) Establish utility services by abstracting from the corresponding alternative service operations mentioned in (2) according to the fact that utility services have nothing to do with specific solutions and are most reusable.

In this case, the only alternative operation left - “send message of ‘order approval has been adopted and input’ to relevant business staff” can be abstracted as utility service of “note”, which includes the operation of “send notice message”.

(8) Adjust and optimize the process logic, and determine the interactive relationship between process services and each member service, treating process service as the centre.

First, recheck process logic to find a more optimal result. For this instance, “enquiry of current inventory” and “enquiry of sales of these signed orders” are two sub-process of “enquiry of product supplies in the enterprise”, and their orders can be changed from sequential execution to parallel execution.

Then, determine the relations between process steps and each member services, and then form the interactive model as shown in Fig.10.

It must be pointed out that there are no absolutely right or wrong results for service modeling. It is closely related to results of preliminary analysis, consideration of technical realization and experience the staff has. Therefore, service models constructed at the service modeling phase are often adjusted in later service design stage. Maybe only a part of the service models can be covered in the actual solutions because of time, cost, technique and other factors.

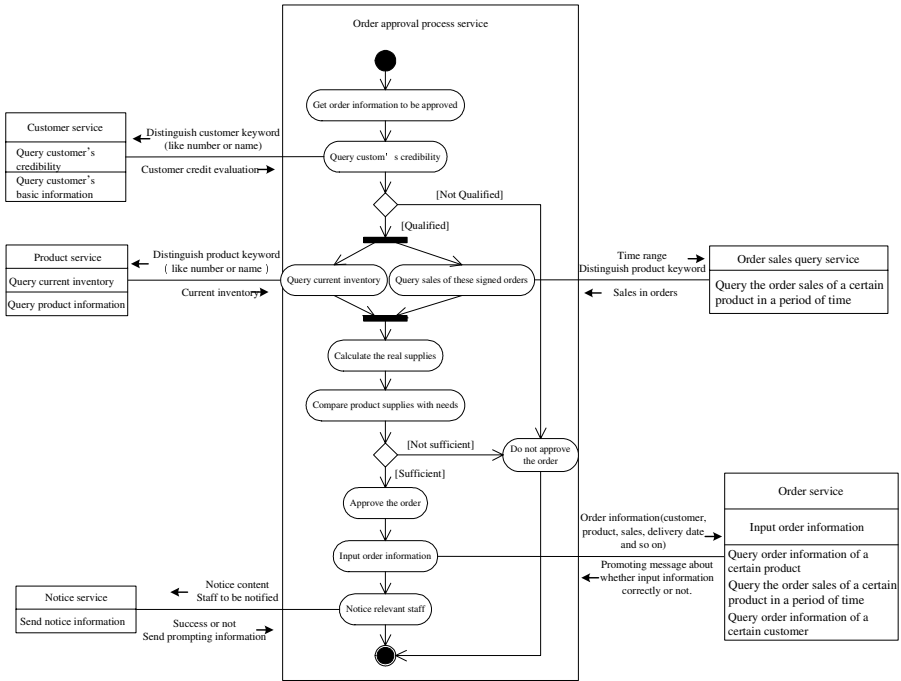


Fig. 10. Interactive relationship model between process services and their member services

We did actual development after adjusting above service models appropriately and designing service interface. We implemented several services in business service layer and application service layer by encapsulating three existing business system with Java program and publishing them on the Tomcat and Axis platform as web services. Then with the help of WS-BPEL tool, we transformed the service models into an executable business process and eventually realized the process automatic processing by compositing member services and design process services and deploy them on the server computer. We designed process services with Oracle’s BPEL tool according to the interactive relationship between process services and their member services shown in Fig.10.

5 Conclusions

Enterprise logic can be expressed by various kinds of enterprise models and be abstracted formally with the help of enterprise modeling. The fundamental purpose of SOA implementation is to provide a loosely coupled enterprise information architecture to satisfy enterprise requirement for automatic operational application and dynamic response of application logic for business logic. As an important step in the SOA implementation process, service modeling should be based on enterprise models to ensure that service models consist with enterprise models and application logic matches with business logic.

A service modeling method based on enterprise models is presented in this paper, which specifies model expressing method and modeling process. By referring to the service hierarchical theory and by recognizing, compositing, adjusting service operation and determining service layer configuration based on enterprise models, we can build a service model centering on process services. Generally speaking, the service modeling method has a good workability and can provide a reference for engineering practice.

Acknowledgement

This paper is sponsored by China National Natural Science Foundation (No. 60474060), Beijing Natural Science Foundation (No.9072007) and China 863 Program (No. 2007AA04Z1A6).

References

1. Yu, J., Han, Y.b.: Service Oriented Computing: Principles and Applications (in Chinese). Tsinghua University Press, Beijing (2006)
2. Erl, T.: Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR, USA (2005)
3. Zimmermann, O., Krogdahl, P., Gee, C.: Elements of Service-Oriented Analysis and Design: An interdisciplinary modeling approach for SOA projects (2004-06-02), <http://www-128.ibm.com/developerworks/library/ws-soad1/>
4. Arsanjani, A.: Service-Oriented Modeling and Architecture (2004-11-09), <http://www-128.ibm.com/developerworks/webservices/library/ws-soa-design1/>
5. Endrei, M., Ang, J., Arsanjani, A., et al.: Patterns: Service-Oriented Architecture and Web Services. USA: IBM International Technical Support Organization (2004)
6. Kim, Y., Yun, H.: An Approach to Modeling Service-Oriented Development Process. In: Proceedings of the 2006 IEEE International Conference on Services Computing (SCC 2006), pp. 273-276. IEEE Computer Society Press, Chicago (2006)
7. Wang, J., Yu, J., Han, Y.: A Service Modeling Approach with Business-Level Reusability and Extensibility. In: Proceedings of the 2005 IEEE International Workshop on Service-Oriented System Engineering (SOSE 2005), pp. 23-28. IEEE Computer Society Press, Beijing (2005)
8. Li, Q., Chen, Y.L.: Global System Design of Enterprise Informationization (in Chinese). Tsinghua University Press, Springer Press, Beijing (2004)
9. Fan, Y.S., Wang, G., Gao, Z.: Introduction to Enterprise Modeling Theory and Methodology (in Chinese). Tsinghua University Press, Springer Press, Beijing (2001)
10. Wang, Y.T., Li, L., Song, H.Z.: Fundamentals and Applications of UML (in Chinese). Tsinghua University Press, Beijing (2006)
11. Xu, B.W., Zhou, Y.M., Lu, H.M.: UML and Software Modeling (in Chinese). Tsinghua University Press, Beijing (2006)
12. BPMI.ORG. Business Process Modeling Notation (BPMN). Version 1.0, Business Process Management Initiative (2004)

Service Oriented Architecture vs. Enterprise Architecture: Competition or Synergy?

Ovidiu Noran and Peter Bernus

Griffith University Australia, School of ICT
{O.Noran, P.Bernus}@griffith.edu.au

Abstract. Currently, Service Oriented Architecture (SOA) is still in its infancy, with no common agreement on its definition or the types and meaning of the artefacts involved in its creation and maintenance. Despite this situation, SOA is sometimes promoted as a parallel initiative, a competitor and perhaps even a successor of Enterprise Architecture (EA). In this paper, several typical SOA artefacts are mapped onto a reference framework commonly used in EA. The results show that the EA framework can express and structure SOA artefacts with minimal or no customisation and can help reason about and establish unambiguous meanings for SOA artefacts across the business. Further on, it is shown how an EA-specific approach can help scope the areas of the business that require attention as a result of the changes brought about by an SOA vision and design principles. This suggests that integrating the SOA effort into the ongoing EA initiative is a best practice that will greatly benefit all business units of the host organisation.

1 Introduction

Although several definitions for Service Oriented Architecture (SOA) exist, the prevalent view appears to be that SOA is in essence an architectural style promoting the concepts of service (packaged business functions with all necessary information) and service consumer as a basis to structure the functionality of an entire business. The SOA concept is not new, originating in the modular, object-oriented and component-based software development paradigms. However, the lack of adequate supporting and realisation infrastructure have in the past hindered its adoption [1]. According to the Gartner Group, after the typical wave of vendor hype and unrealistic expectations, SOA is now recovering from the disillusionment phase and heading towards the plateau of productivity [2]. Even though standardisation attempts are underway, currently there is still no common agreement on a rigorous SOA definition, or the types and meaning of the artefacts that should be involved in the creation and maintenance of an SOA [3]. Furthermore, the realisation that building an SOA involves significant costs and changes to the *entire* business has contributed to SOA being sometimes seen as a separate approach, a competitor and perhaps a successor of Enterprise Architecture (EA) – an increasingly popular approach to describe and manage changes in enterprises so as to enhance their consistency and agility.

Thus, this paper argues that SOA is a style and/or component of EA rather than an alternative or a competitor. This position is supported in two steps. Firstly, the paper shows how a typical EA artefact, namely a reference Architecture Framework (AF), can be used to find common, agreed-upon meanings and actual coverage of the various artefacts involved in an SOA effort. Secondly, it demonstrates how an SOA

endeavour can be analysed from an EA perspective that facilitates a coherent approach across the business units and provides the premise for organisational culture change enabling the lasting success of an SOA project.

2 The Reference Framework

The need to establish a framework early in an SOA project appears to be generally accepted [4-6]. The assumption made in this paper is that if SOA-specific artefacts can be mapped onto an enterprise reference AF in a meaningful way, then the required 'SOA framework' could in fact be a type of enterprise AF, which would support the SOA - EA synergy and integration argument. Thus, several typical artefacts described in SOA literature will be mapped against a reference AF, obtained by combining a number of mainstream Enterprise AFs and validated against several others. Note that a comprehensive mapping of all SOA artefacts currently identified is beyond the proposed scope and space available for this paper; the aim here is to prove the concept and perhaps incite constructive debate.

The reference framework proposed is described in Annex C of ISO15704:2000/ Amd1:2005, and it is called the Generalised Enterprise Reference Architecture and Methodology, or GERAM [7]. ISO15704:2000 sets requirements for reference architectures and methodologies (without prescribing any specific artefacts); GERAM is provided as an example of a generalised enterprise AF that satisfies these requirements. As such, GERAM can be (and has been) used to assess particular AFs, or to establish a selection of AF components to be used in a specific EA project since often, a single AF does not have all the elements required. Several mainstream AFs have been mapped against GERAM [8-10] and a 'Structured Repository' of mainstream AF elements is

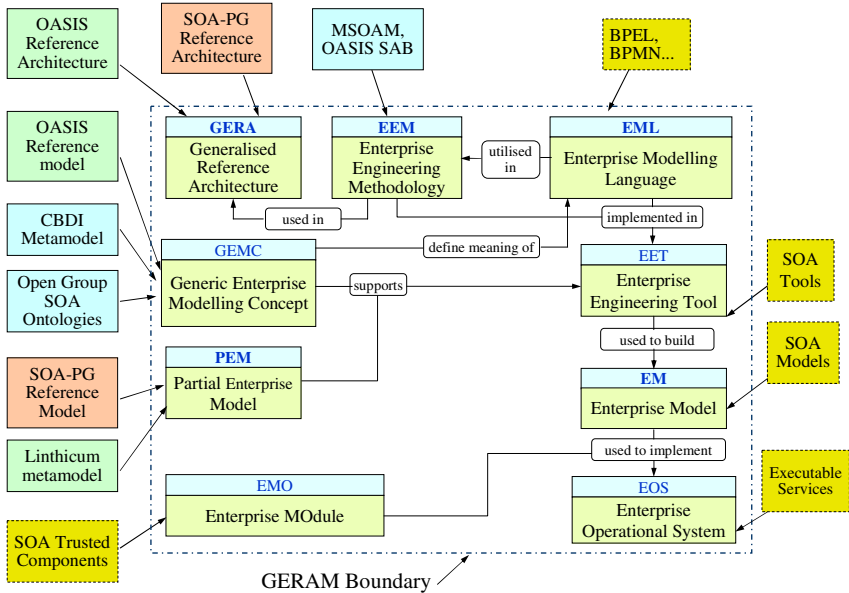


Fig. 1. Sample mapping of SOA artefacts on GERAM (ISO/IEC, 2005)

being built using GERAM as a decomposition and structuring tool [11]. GERAM is one of the most complete reference AFs; in addition, as part of ISO15704:2000, it is regularly reviewed so as to harmonize it with other standardisation efforts such as ISO/IEC 42010:2007 [12]), ISO/IEC 15288:2002 [13], etc. This ensures that GERAM will constantly include a set of essential concepts shared and agreed upon by the EA community.

The Generalised Enterprise Reference Architecture (GERA) component of GERAM contains the multi-dimensional modelling framework (MF) and other essential concepts such as life history and enterprise entity. The GERA MF (see Fig. 2) contains a multitude of aspects that *may* be required in modelling an EA project / product, in the context of the project / product’s life cycle. The GERA MF also features the genericity dimension, which allows representing the meta-models and ontological theories underlying languages used to build partial (e.g. reference) and particular models. Thus, the GERA MF contains *placeholders* for models describing the components shown in the GERAM structure depicted in Fig. 1. Full descriptions of GERAM, GERA and GERA MF are contained in ISO15704:2000 and are beyond the scope and space available for this paper.

3 Mapping Typical SOA Artefacts on the Reference Framework

The following section attempts to map several SOA artefacts currently offered by vendors and / or described in SOA literature, that are deemed of interest to the scope of this paper. The selection of particular artefacts does not imply their endorsement.

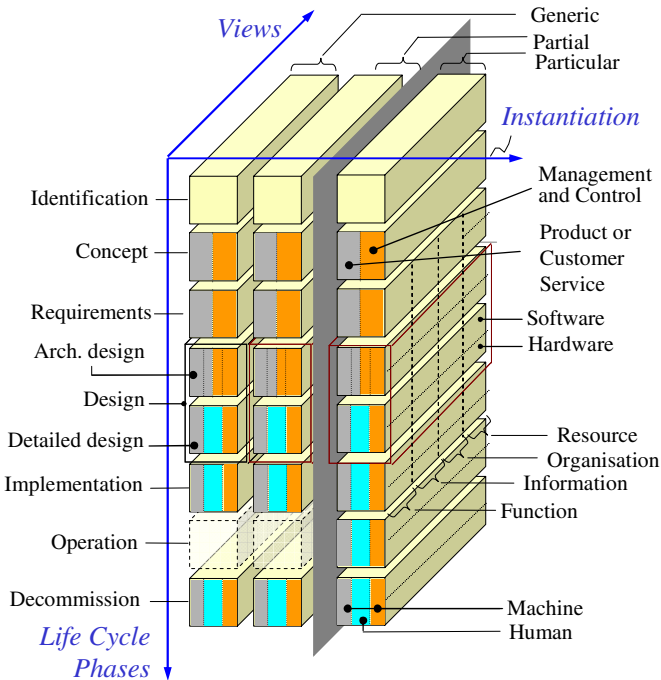


Fig. 2. GERA MF (ISO/IEC, 2005)

3.1 SOA Ontologies

The SOA Working Group (WG) of The Open Group aims to provide ontologies for SOA so as to promote “[...] alignment between the business and information technology communities” [14]. In GERAM, ontological theories are a kind of generic enterprise model, describing the most general aspects of enterprise-related concepts and defining the semantics of the modelling languages used. The Open Group Ontology document currently contains definitions for contract, visibility, registry etc; thus, it maps onto the Generic Concepts area of GERAM (see Fig. 1) and the Generic area of GERA MF (detailed mapping not shown due to space limitations).

3.2 SOA Metamodels

In GERAM, a metamodel describes the properties and relationships of concepts used in the modelling endeavour, as well as some basic constraints, as cardinality [7]. Thus, an SOA metamodel should define relationships between SOA components, list rules for building models and define terminology consistently and unambiguously.

Linthicum [15] proposes an artefact called an SOA metamodel. However, according to the definitions above, the artefact is rather a high-level reference model since it describes an SOA model at the architectural level life cycle phase (see Fig. 1).

Another meta-model proposition is offered by Everware-CBDi [16]. This artefact appears to fulfil the requirements of a meta-model by GERAM (although lacking SOA principles such as loose coupling, autonomy, mediation, etc) and thus can be mapped on the generic concepts area of GERAM. The various artefacts depicted in the metamodel can be mapped onto the aspects of the generic level of the GERA MF.

3.3 SOA Reference Models and Reference Architectures

Many vendors and consultants (IBM, BEA, Oracle, WebMethods, etc) offer what they call ‘reference models’ (RMs) and ‘reference architectures’ (RAs). In GERAM, RMs are seen as blueprints describing features common to specific types of enterprises, while RAs are RMs created at the Architectural Design level.

The OASIS RM [17] in its current version is closer to a meta-model than to an RM from the GERAM perspective since it does not appear to express a blueprint for SOA implementation. OASIS RAs and Patterns do however match the GERAM RA definition since they are RMs for particular SOA systems expressed at the Architectural Design level. The OASIS Concrete Architecture is in EA the Architectural Design level model of a particular SOA system – and thus maps on the Particular level within the GERA MF, at the Architectural Design life cycle phase.

The RA described in the Practitioner’s Guide (PG) authored by Durvasula et al. [18] specifies the structure and the functionality of model components and thus appears to be a proper RM at the Architectural level (RA, according to GERA MF). The proposed mappings of the two artefacts are shown in Fig 1 and Fig. 3.

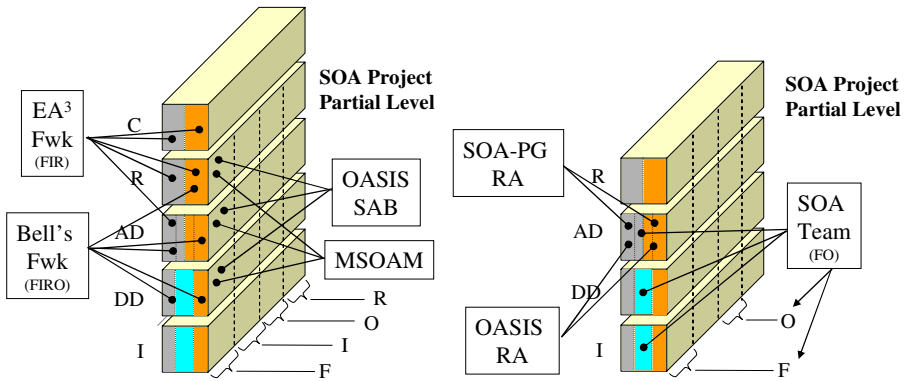


Fig. 3. Sample mappings of MF and methodologies (left) and human aspect of SOA projects (right) on simplified GERA MF (aspects / levels irrelevant to specific mapping omitted)

3.4 SOA Modelling / Documentation Framework

An MF according to ISO15704:2000 is a structure that holds models describing all necessary aspects of enterprises and/or EA projects, along their entire life history. The EA Documentation Framework (DF) is described by Bernard [5] as one of the main components of any EA endeavour. In the SOA domain, McGovern [4] also emphasizes the importance of having a framework guiding the SOA initiative.

It appears that the general meaning given to a DF is in fact that of MF. Knippel [19] describes the SOA DF as a new product, however he suggests investigating whether the SOA and EA frameworks could have common areas and even be merged. This supports the SOA-EA integration proposition made in this paper.

The SOA MF described by Bell [20] provides the conceptual, analysis and logical life cycle phases that may map onto the Requirements, Architectural and Detailed Design phases of GERA; however, the MF appears to lack several other aspects. For example, the human aspect and the management / service distinction are not explicitly represented. Therefore, if such aspects are deemed necessary for the SOA project at hand, elements of other frameworks may need to be employed.

As another example, the EA³ framework described by Bernard [5] as expressed in its graphical form (which may not completely reflect the written content) appears to map on the Partial Level, at Concept and Architectural Design life cycle phases, and cover Function, Information and Resource aspects (see Fig. 3, left).

3.5 Further Mappings of Other SOA Artefacts

Further mappings including SOA life cycle, team, vision, governance, methodologies, quality of service and Enterprise Service Bus have been accomplished in order to support the argument that a reference AF can be used to find common meanings and actual coverage of various artefacts involved in an SOA project.

However, they cannot be shown here due to space limitations. The reader is directed to [21] for details and interpretations of these additional mappings.

4 Defining and Creating an SOA: An EA Approach

From an EA point of view, it is possible to define the SOA concept for a business by extending its present vision to depict the business as a set of reusable services. It is also possible to define SOA design principles as follows:

- technology principles – by declaring service orientation as a technology principle resulting from, and informed by, technology trends analysis;
- information management principles – by mandating common data services;
- organisational principles – by declaring that the business needs to be organised as a set of interrelated and reusable services (this is essentially the SOA principle applied to the entire business);
- organisational and cultural principles – by stating that contribution to reusability should be encouraged, measured and rewarded;
- process principles – by requiring that business processes need to be independent from applications and that business management should be able to own and independently manage / design and roll out changed business processes.

Note that the functional requirements (the ‘tasks’) of the company may not change in terms of what the company does for its customers; however, the *management* requirements do change, in that there are additional, or modified management processes needed to be able to act on the changed principles. The non-functional / resource requirements may also change – e.g. performance requirements (for service *and* management) may have to be stated explicitly (whereupon earlier these were not explicit), because it is known that by adopting the above new principles, QoS can become an essential issue. This is so, because if services become sharable applications they are no longer separately maintained for servicing a dedicated and fixed set of users, and therefore the use of a service by one entity may adversely affect other simultaneous users of the same service in another entity. As a result there are also organisational requirements: there is a need to allocate suitably competent employees to service provision and management in order to ensure the required level of QoS.

A question arises: once the principles and tasks are defined, *should a new requirements specification be created for the entire enterprise?* While possible, this would not be a very efficient course of action. Rather, from the vision and the design principles it is possible to locate the entities that need change and draw a business model that does not need upfront detailed requirements specification. Subsequently, the business model can be used to localise the need for change and to identify the necessary new artefacts (models).

4.1 Sample Business Model of an SOA Scenario

The following example aims to illustrate the previous description of the role of EA artefacts and business model using a modelling formalism derived from the GERA MF.

In the SOA scenario in Fig. 4, the headquarters of a business sets up an SOA project but also the mission, vision, design principles, policies and high-level requirements for the services required. Subsequently, the SOA project starts operating and with assistance from all business units creates the rest of the deliverables required for

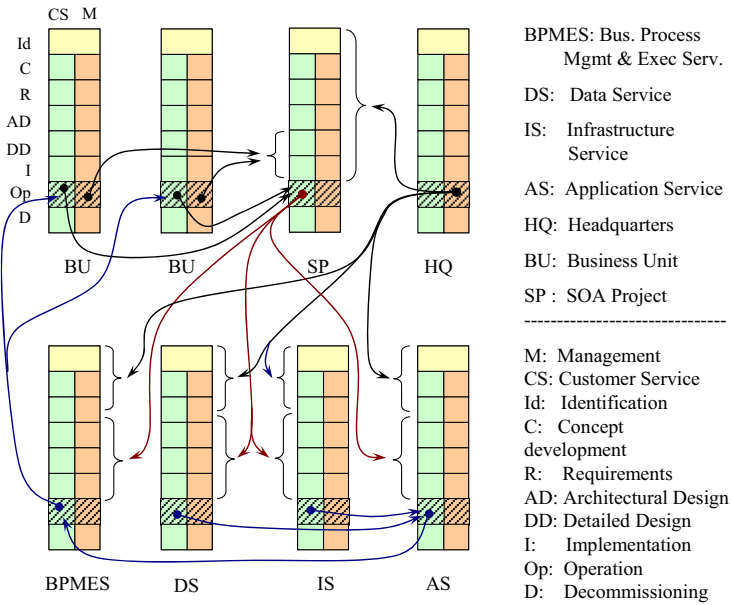


Fig. 4. Relation project / product / services (based on [22])

the business, application and infrastructure services. Once the services are operational, they perform their primary function, i.e. to support the business units' operation. In EA, such representations have proven to be effective in achieving a common understanding of the AS-IS and TO-BE states of the business and scoping the extent of necessary change at each life cycle phase of target entities.

As can be seen from Fig. 4, the business model is in fact an architecture description (a description of the business at the GERA MF architectural design level) that is intended to address a specific stakeholder concern, namely *what* (and *who*) is needed to implement the (SOA) vision that is based on the changed principles. More details are available in [22]; [11] describes a way to use the business model to derive a step-by-step methodology to create and operate the specific SOA project and its deliverables.

5 Conclusions and Further Work

In this paper, we have argued that the use of EA frameworks and approaches is suitable and beneficial in SOA projects. The mappings shown are by no means comprehensive; they rather aim to exemplify how a common reference can help business management and the EA/SOA team work out areas that can be covered by the various artefacts on offer and also point out potential gaps and overlaps. Making sense of the myriad of SOA artefacts created by interest groups, academics, vendors etc is an essential step in gathering stakeholder support for the SOA endeavour. Further on, we have shown how an EA-specific approach can help scope the areas of the business that require attention as a result of the changes brought about by a service-oriented business architecture vision and design principles.

The approach advocated by this paper would promote SOA-EA integration rather than rivalry and be highly beneficial - since EA can help an SOA initiative get off the ground by more accurately identifying and predicting required business and supporting services and sustain it by a cross-departmental approach. EA can also help achieve a cultural change promoting reuse – e.g. by a system of values that rewards business units who share services that become frequently reused.

Clearly, further mappings of SOA artefacts on the reference AF need to be performed in order to increase confidence in the use of EA elements and approaches in SOA projects and perhaps build a repository of EA artefacts most suited to SOA. In addition, the suitability of other EA artefacts such as management maturity models [23] or development kits [24] also need to be tested for use in SOA projects.

References

1. Schönherr, M.: Connecting EAI-Domains via SOA - Central vs. Distributed Approaches to Establish Flexible Architectures. In: Bernus, P., Fox, M., Goossenaerts, J.B.M. (eds.) *Knowledge Sharing in the Integrated Enterprise: Interoperability Strategies for the Enterprise Architect*, pp. 111–113. Kluwer Academic Publishers, Toronto (2004)
2. Fenn, J., Linden, A., Cearley, D.: *Hype Cycle for Emerging Technologies*, Gartner Group (2005) (Retrieved June 5, 2008), from, http://www.gartner.com/teleconferences/attributes/attr_12993_0_115.pdf
3. Erickson, J., Siau, K.: *Web Services, Service Oriented Computing and Service Oriented Architecture: Separating Hype from Reality*. *Journal of Database management* 19(3), 42–54 (2008)
4. McGovern, J.: *Service Oriented Architecture*. In: McGovern, J., et al. (eds.) *A Practical Guide to Enterprise Architecture*, pp. 63–89. Prentice Hall PTR, Upper Saddle River (2003)
5. Bernard, S.A.: *An Introduction To Enterprise Architecture*. AuthorHouse, Bloomington (2005)
6. Sprott, D., Wilkes, L.: *Enterprise Framework for SOA. Component based Development and Integration Journal* (2005)
7. ISO/IEC, Annex C: GERAM, in ISO/IS 15704:2000/Amd1:2005: *Industrial automation systems - Requirements for enterprise-reference architectures and methodologies* (2005)
8. Noran, O.: *An Analysis of the Zachman Framework for Enterprise Architecture from the GERAM perspective*. *IFAC Annual Reviews in Control, Special Edition on Enterprise Integration and Networking* (27), 163–183 (2003)
9. Noran, O.: *An Analytical Mapping of the C4ISR Architecture Framework onto ISO15704 Annex A (GERAM)*. *Computers in Industry* 56(5), 407–427 (2005)
10. Saha, P.: *A Synergistic Assessment of the Federal Enterprise Architecture Framework against GERAM (ISO15704:2000 Annex A)*. In: Saha, P. (ed.) *Enterprise Systems Architecture in Practice*, pp. 1–17. IDEA Group, Hershey (2007)
11. Noran, O.: *Discovering and modelling Enterprise Engineering Project Processes*. In: Saha, P. (ed.) *Enterprise Systems Architecture in Practice*, pp. 39–61. IDEA Group, Hershey (2007)
12. ISO/IEC, ISO/IEC 42010:2007: *Recommended Practice for Architecture Description of Software-Intensive Systems* (2007)

13. ISO/IEC, ISO/IEC15288: Information Technology - Life Cycle Management -System Life Cycle Processes (2002)
14. SOA WG Open SOA Ontology. The Open Group (Retrieved June 23, 2008), <http://www.opengroup.org/projects/soa-ontology/uploads/40/12153/soa-ont-06.pdf>
15. Linthicum, D.S.: SOA Meta-model (PDF). Linthicum Group (Retrieved June 12, 2008), <http://www.linthicumgroup.com/paperspresentations.html>
16. CBDI. CBDI-SAETM Meta Model for SOA Version 2. (Retrieved June 2008), http://www.cbdiforum.com/public/meta_model_v2.php
17. OASIS SOA Reference Model TC. OASIS Reference Model for Service Oriented Architecture V 1.0. OASIS Group (Retrieved June 20 2008), <http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf>
18. Durvasula, S., et al.: SOA Practitioner's Guide. BEA Systems, Inc. (Retrieved April 2008), <http://dev2dev.bea.com/pub/a/2006/09/soa-practitioners-guide.html>
19. Knippel, R.: Service Oriented Enterprise Architecture (Doctoral Thesis). IT-University of Copenhagen, Copenhagen, p. 125 (2005)
20. Bell, M.: Introduction to Service-Oriented Modeling. In: Service-Oriented Modeling (SOA): Service Analysis, Design, and Architecture, Wiley & Sons, Chichester (2008)
21. Noran, O.: Mapping SOA Artefacts onto an Enterprise Reference Architecture Framework. In: 17th International Conference on Information Systems Development - ISD 2008, Cyprus, Paphos (in print, 2008)
22. Bernus, P.: How To Implement SOA For The Whole Of Business. In: Service Oriented Architecture 2008 - Implementing And Measuring Soa Projects To Drive Business Value Sydney, IQPC (2008)
23. GAO. IT - A Framework for Assessing and Improving Enterprise Architecture Management (Version 1.1). US General Accounting Office (Retrieved June 07, 2008), <http://www.gao.gov/new.items/d03584g.pdf>
24. NASCIO. Enterprise Architecture Development Tool-Kit v3.0. National Association of State Chief Information Officers (Retrieved July 2008), <http://www.nascio.org/aboutNASCIO/index.cfm>

SCEP-SOA: An Applicative Architecture to Enhance Interoperability in Multi-site Planning

Karim Ishak, Bernard Archimède, and Philippe Charbonnaud

Université de Toulouse, Ecole Nationale d'Ingénieurs de Tarbes, Laboratoire Génie de Production EA n° 1905, 47, Avenue d'Azereix, 65016 Tarbes, France
{karim.ishak,bernard.archimede,philippe.charbonnaud}@enit.fr

Abstract. In this article, a model of service oriented market is presented. It gives a more equitable opportunity of integration in the business markets for the Small and Medium Enterprises. Nevertheless, multi-site planning is a critical and difficult task due to the heterogeneity between planning applications of the various partners. For this objective, the proposed interoperable and distributed architecture SCEP-SOA for multi-site planning is based on SOA (Service Oriented Architecture), integrating the concepts of the generic model of planning and scheduling SCEP (Supervisor, Customer, Environment, and Producer). This interoperable architecture enables an applicative interoperability as well as semantic interoperability between different planning applications used by the partners.

Keywords: Distributed planning, interoperability, service oriented architecture, electronic marketplaces.

1 Introduction

Most of the Small and Medium Enterprises (SME) have no strong presence in actual production markets because of their financial and technological limits. Today, companies are focusing towards their core competencies and are of this fact unable to produce alone. They elaborate common projects and participate in one or several networks of subcontractors, co-contractors, customers and suppliers. Companies create Supply Chains (SC) [1] to satisfy the needs of the market. The Supply Chains are mainly dominated by the big companies or donors of orders who resort today to project oriented marketplaces for improving their competitiveness.

Project oriented marketplaces have for objectives a better description of the various lots of a project as well as a better communication with the partners and a synchronization of the activities. Nevertheless, the realization of lots requires high skills and important technical and economical means, what excludes most of the SMEs [2]. Numerous solutions were proposed to enhance the position of the SMEs on the production market [2][3]. However, these methodologies have numerous limits, notably the short-term incapacity to attract a large number of buyers, sellers and products, and also the problem of trust between the partners.

The motivation of the proposed service oriented market's model is to allow the SME freeing itself from economical and technological requirements of the donors of

orders existing in project oriented markets, by offering an independent and fair integration to markets. Nevertheless, multi-site planning is a critical and difficult task due to the heterogeneity between planning applications of the different partners. To solve this heterogeneity, a distributed and interoperable planning and scheduling architecture is presented. In section 2, the project oriented markets and their limits as well as the service oriented markets and their characteristics are presented. In section 3, the generic multi-agent model for multi-site planning and its need for interoperability are briefly described. Section 4 presents the adopted reference model for interoperability. In section 5, the proposed architecture SCEP-SOA for planning and scheduling integrating the reference model for interoperability and the agent's technology is detailed. Section 6 handles the semantic interoperability in the proposed architecture. Section 7 collects the points of interest of the proposed interoperable architecture and presents the future works.

2 Project Oriented Markets Versus Service Oriented Markets

In the world of the electronic marketplaces [4], the big companies (donors of orders) propose their own project oriented marketplaces where they describe their projects which they wish to achieve by defining all the lots and specifying for each of them the required delay, the necessary skills and the standards to be respected. Subcontractor companies make for the lots which interest them and for which they have skills, propositions in terms of cost, lead time, end date, and by the respect for the standards. The selection of the partners for the realization of a project is made by the donor of orders who often sets up a mechanism of top-down bids for companies respecting the fixed standards. A Supply Chain will be created with subcontractors companies having for administrator the donor of orders. Nevertheless, project oriented marketplaces are not convenient for the SME which cannot satisfy the constraints of integration imposed by the donor of orders in terms of capacities, skills and standards. That excludes the SME owning the required skills but not the human and financial capacities to successfully produce lots concerning them, especially when they have to assume a very high technological cost for upgrading their communication and planning systems.

Unlike project oriented marketplaces, service oriented markets put in relation several customers with several producers or suppliers of services. A service can be a simple action, a set of actions or a composite service built by means of elementary services. *A priori*, no relation binds the services together. Once the services are defined in the market, the customers, wishing to achieve some projects, come on this market to negotiate the realization of their projects in terms of services. The management of the marketplace is realized by an entity independent from suppliers and customers. The declaration of the know-how and the initialization of projects are essential functions of this type of markets. The declaration of the know-how concerns the publication of the services that can be offered by the producers. A producer can offer one or several services. A service can be supplied by one or several producers. The initialization consists in describing every project by means of a selection of services available on the market. A project P requires one or several services. Several projects can require the same service. It is not possible to declare a project on the marketplace if the

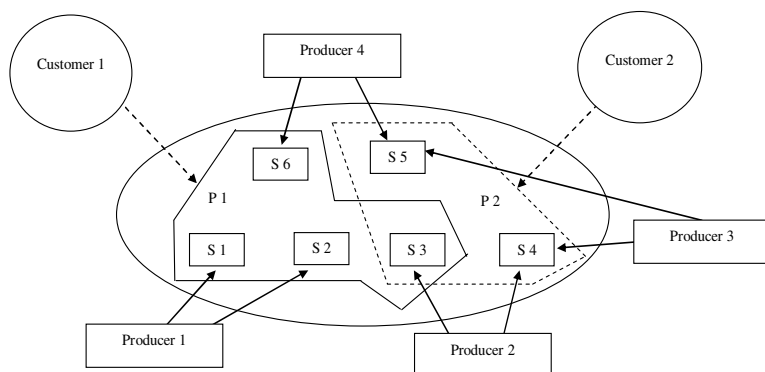


Fig. 1. Declaration of know-how and initialization of projects in service oriented market

services required for all his lots are not all available. Fig. 1 illustrates these two functions (declaration of know-how and initialization of projects) in a service oriented market.

It is shown that a service can participate in several projects (case of P1 and P2 for S3) and can be supplied by several producers (case of S4 and S5). Service oriented markets reduce the barriers of integration met by the SME in project oriented markets and offer fair opportunities of integration by reducing the strong requirements of the donors of orders. Every customer introduces and manages its projects independently of the other customers and producers. Therefore, every project activates a SC which includes the customer and its various producers. The producers can thus participate simultaneously in several SCs. The presence of the several SCs in the market requires a multi-site planning between the partners of the same SC and the coordination of activities between the various SCs, because of the participation of a supplier in several SCs.

3 Towards an Interoperable Planning Model in Service Oriented Markets

The Multi-Agent Systems [5] offer a simple framework to model the various components of a production system. Agents facilitate a natural distribution of the decision, obtaining a scheduling from the local behaviour of the agents and facilitate the ability to react [6]. Several planning approaches based on agent technologies were developed [7][8]. Most of these approaches are based on the communication protocol Contract-Net [9]. In Fig. 2, the conceptual model of planning SCEP (Supervisor, Customer, Environment, and Producer) is described. It proposes a distributed scheduling approach, more detailed in [10], which offers cooperation between customer agents and producer agents via a shared environment (blackboard) and under the control of a supervisor agent. The supervisor agent creates, at the beginning of the process, the customer agents and the producer agents.

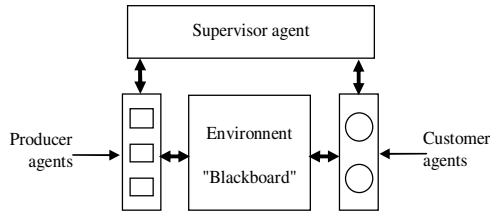


Fig. 2. SCEP conceptual model

The scheduling is obtained after a finished number of cycles each one corresponds to the activation of the customer agents followed by that of the producer agents. Every customer agent makes a planning with infinite capacity of the tasks of his projects then puts down in the SCEP environment, independently of the other customer agents, for each of these tasks, an offer where are indicated his wishes in terms of temporal placement and activity. The producer agents make a planning with finite capacity for the tasks recovering from their domains of skill and make for each of them bids by indicating the temporal placement where the task will be carried out as well as its cost of treatment. This methodology is similar to the functioning of service oriented market presented in the previous section. The mechanism of cooperation, set up in SCEP, uses principles close to the Contract-Net protocol. The necessity of managing better the multi-site activities and the development of the applications and the distributed systems had for consequence the design of new architectures based on multiple networked SCEPs. To elaborate multi-site plans in a distributed way, the SCEP model is enriched by a communication module and by ambassador agents to allow the connection via CORBA between various SCEP models. Although the SCEP model solves in a distributed way the scheduling problem, it presents some limits for its deployment in a service oriented market. It is due to the fact that a preliminary localization of the various sites is necessary in the application of this model as well as no independence between the methods of planning used on the various sites is authorized. Heterogeneity between the partners as well as the lack of integration of standards highlights the lack of interoperability.

4 Reference Model for Interoperability

Interoperability is the possibility for systems or components to exchange information and to be able to use exchanged information [11]. Interoperability is reached only if the interaction between two systems can be realized on data, resources and business process levels with semantics defines in the business context [12]. The main key of interoperability is the respect of standards.

Service Oriented Architecture (SOA) is the reference model for interoperability. It allows gathering the enterprises' software applications in interoperable and reusable services. The principal aim is to authorize the applications to communicate and work together, whatever their respective platform. Interoperability is carried out by means of services. These are components which interfaces and contracts of use are known, and independent of any system. The main actors occurring in SOA are the consumer

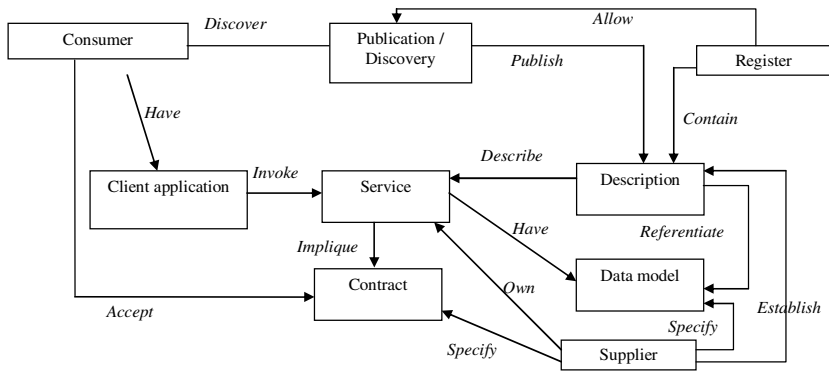


Fig. 3. Key concepts of SOA and their relations

which is the entity that uses the service, the supplier which is the entity that owns the service, and the register which contains the description of the services. Each service is autonomous, independent of other services, and can be discovered and invoked in a dynamic way. There are not official specifications of SOA. Nevertheless, the principal federator and key concepts and their relations are shown in Fig. 3 [13].

The service is a function encapsulated in a component which can be requested by a query composed by one or more parameters and providing one or several answers. The description of the service concerns the description of the input parameters of the service, the format and the type of the output data. The mechanism of service publication publishes in a register the services available to the users, while the mechanism of service discovery concerns the possibility of seeking a service among those which were published. The service invocation represents the connection and the interaction of the customer with the service. The contract of service is a specification of the interaction between the consumer and the supplier of the service. It specifies the format of the service request and answer; it can also specify the levels of quality of service.

5 Interoperable Architecture SCEP-SOA

In this service oriented context, architecture of distributed and interoperable planning is proposed. It integrates the concepts of the reference model SOA and the generic model of planning SCEP. Beyond putting in relation customers and suppliers, this architecture masks for the customers and the suppliers the complexity of the network of planning applications to be set up to carry out the projects. The proposed architecture SCEP-SOA is presented as a service oriented marketplace dedicated to multi-site planning and connecting customers or consumers of services and producers or suppliers of services via a register. Each supplier describes its know-how by gathering all its activities in a composite service. For example, in the field of the production planning, a service can be an activity of turning, milling, etc. or can gather a whole of these activities. The supplier defines in a service description the functional (input and

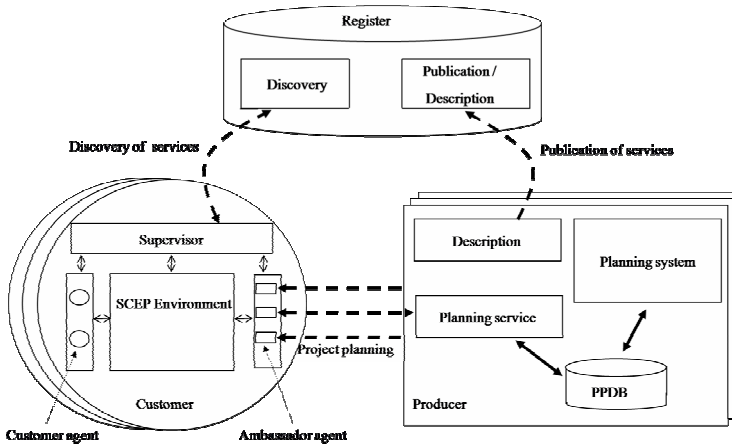


Fig. 4. Interoperable architecture SCEP-SOA for multi-site planning

output parameters, format of the request/answer, etc.) and non-functional properties (service activities, service location, provider address, general description, etc.). In Fig. 4, the functioning of the proposed architecture is shown and the mechanisms of integration of the SOA reference model concepts and the generic model of planning SCEP is displayed.

Each supplier has a planning system in order to plan the offered activities and a Projects and Plans Data Base (PPDB) to stock the clients' project and the scheduling plans proposed for these projects. After describing its activities in terms of service, the producer publishes the service description in the register. From the description of their projects in terms of activities desired for the realization of the necessary tasks, the customers launch a request to the register in order to discover the services able to carry out these activities as well as a pre-selection of the suppliers. The invocation of the services and the exchange of information between the register and the other actors are done by the means of the Simple Object Access Protocol (SOAP). Following the pre-selection of the suppliers able to carry out his projects, the customer sets up in its environment a SCEP model. The environment of this SCEP model will connect customer agents associated with the concerned projects and ambassador agents representing the planning systems with the pre-selected suppliers. The planning service stocks in the PPDB the projects received from ambassador agents of the customers. After the scheduling phase, the service gets back from the PPDB the scheduling plans of the different customer projects and responds to the ambassador agents of corresponding customers. Then, the ambassador agent puts the scheduled plans in the SCEP environment to be discovered and treated by the customer agents. Note that the activities which the suppliers must carry out can be decomposed into sub-activities and reintroduced in a SCEP-SOA marketplace in order to find the suppliers to carry them out.

Each customer is at the origin of a supply chain gathering the customer, its producers, the producers of the producers, etc. The development of a multi-site planning requires, from every producer, the definition of the frequency with which it will start again its planning system. At the producer side, if a request of planning service has occurred during a planning cycle, this request will be taken into account by the

planning system in the next planning cycle. For this reason, the customers should be informed about the maximum waiting time of the service response. This information must be specified in the contracts of concerned services.

This architecture implements the various concepts set up in the service oriented markets and permits applicative interoperability thanks to SOA model which enables communication and interaction between various heterogeneous planning systems independently from the platforms and infrastructures of these systems. This architecture carries out a communication and cooperation between the different planning systems of the partners by offering a loose coupling between the services, a functioning independent of the platform of implementation, a possibility for re-using of the services, and better upgrading capabilities. According to the Enterprise Interoperability Framework [14], the proposed SCEP-SOA architecture overcomes technological barriers that result from the incompatibility of the information technologies (architectures, platforms, infrastructures, etc.) used by each partner in the planning process. SCEP-SOA concerns data, service and process levels because it offers an interoperable solution which allows sharing data and planning information coming from heterogeneous planning systems, it makes function together various planning applications designed and implemented independently, and it allows various planning process to work together without modifying considerably the internal planning process of each partner. Concerning the interoperability approaches defined in this Enterprise Interoperability Framework, SCEP-SOA is considered as a federated interoperability approach due to the fact that no partner imposes its models, technologies, languages or methods of work to other partners.

6 Semantic Interoperability in SCEP-SOA

Although the interoperability between the planning applications is assured thanks to SOA, these applications are built by different designers which do not use the same vocabulary of planning. That leads to a problem of semantic interoperability between the partners. To insure the good understanding and interpretation of the information exchanged between these applications, the proposed solutions are very often based on ontologies [15][16]. These are considered as semantic interoperability tools which allow modelling of the planning concepts manipulated by the various planning applications in a formal way. Each partner has its own ontology leading to a problem of ontology heterogeneity. To solve the heterogeneity problem between these ontologies and insure semantic interoperability, the proposed strategy consists in setting up a global and common ontology which will serve as *interlingua* in the exchange of the information between the various planning applications. Mechanisms of ontology transformation [17] are needed by customers and producers to transform the received data described according to the global ontology in data expressed according to the local ontology and *vice versa*. The first step of this strategy concerns the discovery and the representation of ontology mappings between local planning ontology and global ontology. In Fig. 5, components added to SCEP-SOA in the first step of the semantic strategy are shown.

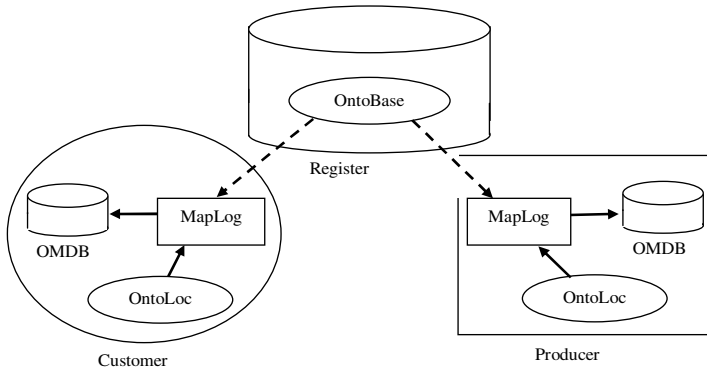


Fig. 5. Semantic components to discover and represent ontology mappings

OntoBase represents the global common ontology used as reference ontology in the exchange of information between planning applications. This ontology is set up by the register and is imported in MapLog during the phases of service publication and service discovery. MapLog is a representative name of the component used to discover and represent the ontology mappings. Many tools are developed to establish mappings between two ontologies. Each partner can use a different mapping tool to establish the ontology mappings between its local ontology (OntoLoc) and the global planning ontology (OntoBase). The ontology mappings are then stored in an Ontology Mappings Data Base (OMDB) to reason with them in the planning phase. Note that ontology translation mechanisms [18] are needed if the local ontology of a partner is described with an ontology language different from the OntoBase's one.

The second step of the proposed semantic strategy, depicted in Fig. 6, concerns the reasoning about the ontology mappings established in the first step. A transformation agent is set up at the customer and producer sides to transform received information from global ontology to local one and *vice versa*. At the customer side, the ontology mappings are implemented in a Customer Transformation Agent (CTA) to transform information about client projects described in SCEP ontology to projects described according to the global planning ontology. After receiving the transformed information from the CTA, the ambassador agent invokes the Planning Service (PS) of the corresponding producer. At the producer side, a Producer Transformation Agent (PTA) is set up to reason with ontology mappings. It transforms information received by the service about customer projects described according to the global ontology in information according to the local producer ontology.

After transformation, these projects are stored by the PS in the PPDB to be planned later. After the planning was achieved, the PS gets from the PPDB the proposed plans which are described according to the local planning ontology of the producer. The PS asks the PTA for transforming these plans into plans described according to the global ontology and sends them to the corresponding ambassador agent. Receiving the response from the PS, the ambassador agent asks the CTA for transforming these planning results into plans described according to the SCEP ontology, to be treated correctly by the customer agents.

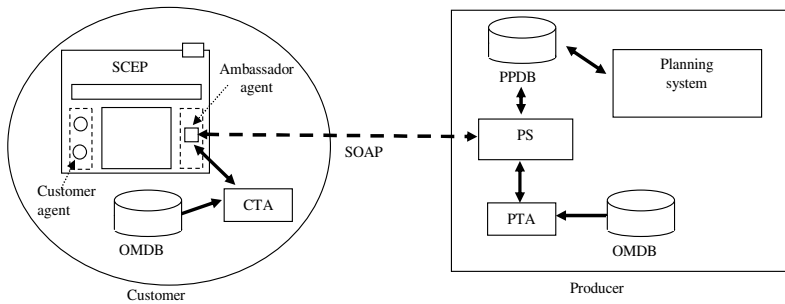


Fig. 6. Transformation agents added to insure semantic interoperability between planning ontologies

This strategy offers a standardization of the planning information exchanged between partners and enables a semantic convergence thanks to the global planning ontology, what insures well understanding and well interpretation of the planning information exchanged between heterogeneous planning systems. In addition, this strategy reduces the complexity of the ontology mappings at the partners' level which is limited to mappings only with the global planning ontology.

7 Conclusion

A distributed architecture SCEP-SOA was described to enable an interoperable planning process in service oriented markets. It solves the heterogeneity between different planning systems and enables a multi-site planning in an independent way. This architecture is based on the SCEP model for elaborating multi-site plans and the reference model SOA to make it possible to localize and interoperate with other different planning systems. Ontologies are used to capture the semantic of the planning concepts manipulated by the different partners. Additional components are added to the SCEP-SOA architecture to insure semantic interoperability between these different planning ontologies. The adopted solution for semantic interoperability is based on a global ontology describing the information exchanged between various planning systems, as well as the ontology mappings to transform information between the global ontology and local planning ontologies. Future works focus on the building of the global planning ontology and on the implementation of this architecture as well as its performance evaluation.

References

1. Pires, R.I.S., Bremer, C.F., De Santa Eulalia, L.A., Goulart, C.P.: Supply Chain and Virtual Enterprises: Comparisons, Migration and a Case Study. *International journal of Logistics: Research and applications* 4, 297–311 (2001)
2. Udomleartprasert, P., Jungthirapanich, C., Sommechai, C.: Supply Chain Management – SME Approach. *Managing Technologically Driven Organizations: The Human Side of Innovation and Change* 2, 345–349 (2003)

3. Park, J., Yang, J.: An International SME E-Marketplace Networking Model. In: *The economics of Online Markets and ICT Networks*, pp. 245–257. Physica-Verlag HD (2006)
4. Eurochambres, Association of European Chambers of Commerce and Industry: *Opportunities and Barriers for SMEs - A First Assessment*. Draft Working Document of DG Enterprise B2B Internet Platforms (2002)
5. Ferber, J.: *Multi-agent systems - an introduction to distributed artificial intelligence*. Addison Wesley Longman (1999)
6. Jiao, J.R., You, X., Kumar, A.: An Agent-Based Framework for Collaborative Negotiation in the Global Manufacturing Supply Chain Network. *Robotics and Computer-Integrated Manufacturing* 22, 223–255 (2006)
7. Lima, R.M., Sousa, R.M., Martins, P.J.: Distributed Production Planning and Control Agent-Based System. *International Journal of Production Research* 44, 3693–3709 (2006)
8. Shen, W., Wang, L., Hao, Q.: Agent-Based Distributed Manufacturing Process Planning and Scheduling: A State-of-the-Art Survey. *IEEE Trans on systems, man and cybernetics* 36, 563–577 (2006)
9. Smith, R.: The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Trans on Comp.* 29, 1104–1113 (1999)
10. Archimède, B., Coudert, T.: Reactive scheduling using a multi-agent model: the SCEP framework. *Engineering Applications of Artificial Intelligence* 14, 667–683 (2001)
11. IEEE: *IEEE: Standard Computer Dictionary. A Compilation of IEEE Standard Computer Glossaries* (1990)
12. Chen, D., Doumeingts, G.: European initiatives to develop interoperability of enterprise applications-basic concepts, framework and roadmap. *Annual Reviews in Control* 27, 153–162 (2003)
13. Nickull, D.: *Service Oriented Architecture*. Adobe System Incorporated (2005)
14. Chen, D., Dassiti, M., Elvesaeter, B.: *Enterprise Interoperability Framework and knowledge corpus*. Research report of INTEROP NoE, FP6 Network of Excellence, Contract n 508011, Deliverable DI.2 (2006)
15. Maedche, A.: Ontology Learning for the Semantic Web. *IEEE Intelligent Systems* 16, 72–79 (2001)
16. Obitko, M., Marik, V.: Ontologies for Multi-Agent Systems in Manufacturing Domain. In: *13th International Workshop Database and Expert Systems Applications*, pp. 597–602 (2002)
17. De Bruijn, J., Ehrig, M., Feier, C., Martin-Recuerda, F., Scharffe, F., Weiten, M.: *Ontology mediation, merging and aligning*. In: *Semantic Web Technologies*, Wiley, Chichester (2006)
18. Dou, D., McDermott, D., Qi, P.: Ontology Translation on the Semantic Web. In: Spaccapietra, S., Bertino, E., Jajodia, S., King, R., McLeod, D., Orłowska, M.E., Strous, L. (eds.) *Journal on Data Semantics II*. LNCS, vol. 3360, pp. 35–57. Springer, Heidelberg (2005)

IWSSA 2008 PC Co-chairs' Message

The development of software systems is of vital concern to industry, society, and government. However, quality software systems will be possible only if their architectures are designed in a manner to accommodate and satisfy certain requirements. Designing such software architectures mandates the understanding of the high-level pictures of the environment in which the developed software will be used, under a variety of usage scenarios. In this regard, the papers that appear in these proceedings address innovative techniques, methodologies and processes to design software architectures, which are all intended to meet requirements and constraints one way or another. In particular, the papers cover important quality attributes of software systems ranging from users' requirements (security, reliability, usability) to developers' requirements (reusability, maintainability). These papers highlight what appears to be a clear trend in recent technologies (frameworks, ontologies, aspects, services, etc.) towards developing quality architectures from the viewpoint of the various phases in the software lifecycle, namely: requirements modeling, design, prototyping, implementation and evaluation, etc.

Most papers include both theoretical contents and real-case studies for systems with varying scopes and applications, such as enterprise, control, pervasive computing and collaboration. In all probability, they will contribute to a more successful workshop with enriching discussions. We sincerely hope that the workshop will be an appropriate forum where the sharing of knowledge and experiences about system/software architectures promotes new advances in both research and development. We also hope that readers can enjoy the papers in the proceedings.

This will be the 7th International Workshop on System/Software Architectures (IWSSA 2008), and the first time it has joined with the On The Move Federated Conferences and Workshops (OTM 2008). As was the case for previous IWSSA workshops, the present edition received an excellent response and a total of 23 submissions out of which 14 were selected as full papers and 2 as short papers. The rest of the papers also deserve special mention because they were well rated. This time again, authors of selected quality papers, from those presented at IWSSA 2008, will be invited to submit significantly extended versions to a special issue of the Journal of Science of Computer Programming (Elsevier).

We would like to sincerely thank all of the chairs for their constant help and support, the Program Committee members for their excellent work, and most importantly the authors of the papers for their very interesting and high-quality contributions, which make possible this workshop series every year.

November 2008

Lawrence Chung
José Luis Garrido
Nary Subramanian
Manuel Noguera

Semantic-Aided Interactive Identification of Reusable NFR Knowledge Fragments

Claudia López¹, Hernán Astudillo¹, and Luiz Marcio Cysneiros²

¹ Universidad Técnica Federico Santa María, Valparaíso, Chile

² York University, Toronto, Canada

Abstract. Understanding Non-Functional Requirements (NFRs) and trade-offs among them is a key task for systems architects. Modeling notations have been proposed to represent NFRs and tradeoffs among them, yet identification of relevant knowledge inside models (for understanding and/or reuse) remains quite simplistic and manual. This paper proposes to address fragment identification as a problem best served with interactive aids and presents a faceted exploration approach to explore NFR solutions and identify reusable model fragments. NFRs and trade-offs are represented as ontologies, thus opening the door to model merging and high-end visualization. The approach is illustrated with a real-world model, and a prototype tool is introduced. The ultimate goal of this effort is enabling reuse of NFR and trade-off knowledge.

Keywords: Software Architecture, NFR, Faceted Search, Ontology.

1 Introduction

Architecture is critical to the realization of many qualities of interest in a system and these qualities should be designed and should be evaluated at the architectural level [1]. Each quality, also known as Non-Functional Requirement (NFR), may call for different possible alternatives to implement this quality in the software. Moreover, one alternative will frequently conflict with another leading the developer to tradeoff between alternatives.

Recording architecture decisions and rationale helps to understand **how** the architecture is addressing the required qualities and **why** these decisions have been chosen among the alternatives. The rationale plays an important role since decisions will vary according to the problem being addressed at the time. Having the rationale helps one to better understand how each alternative will contribute to address an NFR to the problem at hand. Moreover, sharing or reuse this knowledge with each design experience eventually augmenting it. An interesting research problem is how to support recording and reuse of NFR knowledge in systematic, ideally using automated ways.

To support reuse of NFR knowledge, Chung et al. proposed “NFR catalogues” to store NFRs and their known design solutions represented as Softgoal Interdependency Graphs (SIGs) [2]. Cysneiros et al. [3] proposed the use of a softgoal network indexed by a list of NFR types. As Cysneiros stated [4], these catalogues were proposed without any organizational support towards facilitating

reuse. Later work [4] proposed softgoal networks stored in a relational data model and indexed by a faceted schema.

This article presents an ontology-based configurable faceted search to identify valuable knowledge fragments. NFR knowledge is reified as ontology instances. Facets and their values are generated using concepts and relationships of the ontology. Ontology fragment templates are also defined to represent relevant knowledge fragments inside NFR models. Faceted and multi-faceted search are executed by SPARQL queries. The search outputs can be organized by valuable knowledge fragments. Architects can choose which kind of classification criteria and knowledge fragments they are interested in. A prototype implementation realizes these ideas and preliminary results are shown in this article. This semantic-based approach aims to organize and explore NFR knowledge to find out promising reuse candidates.

Section 2 introduces a motivation. Section 3 surveys current NFR catalogs and ontology-based approaches to describe NFR knowledge. Section 4 explains Semantic Web concepts and describes the proposed approach. Section 5 shows a prototype implementation and Section 6 presents discussions and conclusions.

2 Motivation

Reusing NFR knowledge requires discovering related past design solutions or relevant knowledge inside solutions. However, searching information within large knowledge stores can be quite daunting. The search criteria and relevant knowledge fragments can, and usually are different for different users. We visualize some search scenarios:

Search past solutions related to (multiple) NFRs: The architect searches past solutions that satisfy the same set of NFRs of current interest.

Search interdependencies: The architect wants to discover interdependencies as yet unknown to him to include them in his evaluation process.

Search conflicts related to an NFR: The architect wants to find out arguments related to specific conflicts to solve similar conflicts currently at hand.

Search past solutions in an specific domain: The architect looks for past solutions in the same or similar domains.

Unfortunately, current NFR catalogues [2,3] have fixed search criteria and their results are an entire softgoal network. Our ontology-based approach aims to explore NFR knowledge more intuitively depending on the information the user actually requires.

3 Related Work

Several strands of work are pertinent to the problem at hand.

3.1 Softgoal Interdependency Graphs (SIGs)

Chung et al. introduced SIGs [2] to explicitly describe design decisions and rationales to achieve NFRs. SIGs are based in the *Softgoal* concept. Softgoals may be addressed by different possible alternatives (operationalizations). However, most of these alternatives will only satisfy this softgoal within acceptable limits, hence, the term "satisfice" coined by Simon [5].

NFRs can be represented as softgoals in a SIG: an *NFR softgoal* acts as overall constraints or requirements on the system that might conflict with another NFR softgoal. An *operationalizing softgoal* models a design or implementation component that satisfices the NFR Softgoal. A *claim softgoal* is a mean to add argumentation rationale. Each Softgoal have attributes such as *priority* to represent its criticality, *type* that refers to the related NFR and *topic* that defines the softgoal's scope.

Figure 1 shows a SIG that partially models a Credit Card System [2]. Accuracy of the account information and Response Time for storing and updating that information were considered critical softgoals. They were further refined. At the end, they could be operationalized in different ways Perform First [GoldAccount.highSpending], Auditing [GoldAccount.highSpending] or Validation [GoldAccount.highSpending].

Two softgoals can be inter-related using an *Interdependency*. In the downward direction, the parent softgoal is refined and produces one or more offspring softgoals. Parents and offspring are related by interdependencies. In the upward direction, offspring softgoals make contributions, positively or negatively, to the

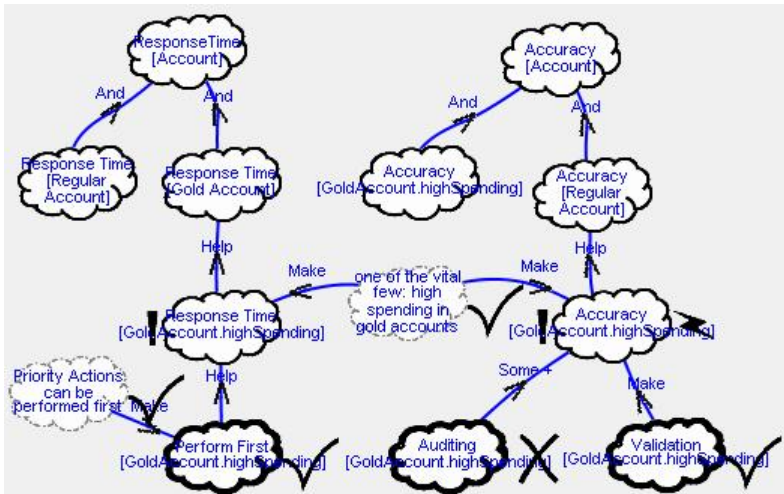


Fig. 1. Example: Partial SIG - Health Insurance System

satisficing of the parent softgoal. *Contributions* can be labeled as Break, Hurt, Unknown, Help, Make, Some-, Some+, Or, And, or Equal. There are three types of *refinements*:

- A *decomposition* is a refinement between two softgoals of the same kind.
- An *operationalization* refines NFR softgoals to operationalizing softgoals.
- An *argumentation* uses claims to explain softgoals or interdependencies.

Correlations capture knowledge about generic interactions between softgoals. This knowledge may have been collected from the literature and/or industrial experience. They are frequently inferred from correlation rules when developers are analyzing conflicts and harmonies or when changes have been introduced. Correlations are not usually explicitly defined by the developer.

The evaluation procedure determines the degree to which NFRs are achieved by the design decisions. The satisfaction degree is represented with a Label and its value can be Satisficed, Weakly Satisficed, Undecided, Weakly Denied, Denied, or Conflict. In Figure 1, `Perform First [GoldAccount.highSpending]` and `Validation [GoldAccount.highSpending]` were evaluated Satisficed and `Auditing [GoldAccount.highSpending]` was considered Denied.

3.2 NFR Catalogues

Chung et al. [2] proposed to store NFR knowledge into three catalogues that are organized according to fixed criteria:

NFR Types catalogue organizes NFRs as a large taxonomy list or a tree according to their subtypes.

Correlations catalogue sorts operationalizing softgoals depending on their contribution to achieve NFR types.

Methods catalogue organizes refinements according to their type. Decompositions and operationalization are further sorted as NFR-specific, generic, or developer-defined methods. Argumentations are also categorized according to their consultation source.

Cysneiros et al. [3] proposed to use a softgoal network indexed by a list of NFR types. Although later he has stated [4] that these catalogues were proposed without any organizational support towards facilitating reuse. Fixed classification does not help to reuse the indexed knowledge intuitively and it can hamper the catalogue usability.

Later, Cysneiros et al. [4] proposed softgoal networks stored in a relational data model and indexed by a faceted schema. It used four facets to organize the knowledge: type, list of related types, list of operationalizations, and topic.

Our approach provides a more flexible faceted search and NFR knowledge recovering by using ontology-based storage. The remainder of this section surveys current ontology-based descriptions of NFR knowledge.

3.3 Ontology-Based Description of NFRs and Architecture Rationale

Few semantic-based approaches have been proposed in the NFRs context. ElicitO [6] provides an ontology-based knowledge about NFR requirements and related metrics to support NFRs elicitation. Dobson et al. [7] presented a domain-independent ontology for NFRs and their metrics. These works do not support NFR solution descriptions and hence do not help to deal with them.

Kruchten et al. [8,9] proposed an ontology of design decisions and relationships among them. Akerman and Tyree [10] introduced an ontology to describe software architectures. These works do not support an evaluation process and correlations therefore they do not help to deal with trade-offs.

Sancho et al. [11] proposed to use an ontological database to describe SIGs and exemplified its use with the Software Performance Engineering Body of Knowledge. Their proposal consists of two ontologies: the NFR ontology and the SIG ontology. The former describes the NFR concepts and their relationships; the latter depicts SIG constructs and their relationships. We have identified two shortcomings of SIG ontology: It does not define any class to describe the **Correlation** interdependency and it does not enforce the use of the proper kind of **Softgoals** as parent and offspring of each **Refinement**.

4 Ontology-Based Faceted Search and Fragment Identification of NFR Knowledge

This section describes a flexible ontology-based faceted classification and ontology fragment templates to retrieve valuable NFR knowledge fragments. Multi-faceted search can help to intuitively find out reusable past solutions and knowledge fragments are used to organize search results according to the user's needs.

4.1 NDR Ontology: NFRs and Design Rationale Ontology

We have developed the NFRs and Design Rationale (NDR) ontology to describe well-formed SIG models. NDR ontology is written in OWL [12]. Classes represent SIGs concepts. Properties describe feasible relationships among concepts. Inference rules are also defined to highlight designer's decisions in the evaluation process such as conflict resolution. The ontology classes and properties are shown in the Figure 2. Details of NDR ontology can be found in [13, 1].

4.2 Ontology-Based Faceted Search

Ranganathan introduced faceted classification [14] as a concept-based classification. Facets are groups of simple concepts that allow describing the objects in the knowledge store. Object descriptions are then synthesized from combinations

¹ Details of inference rules have been submitted to the Requirements Engineering track of ACM Symposium on Applied Computing 2009.

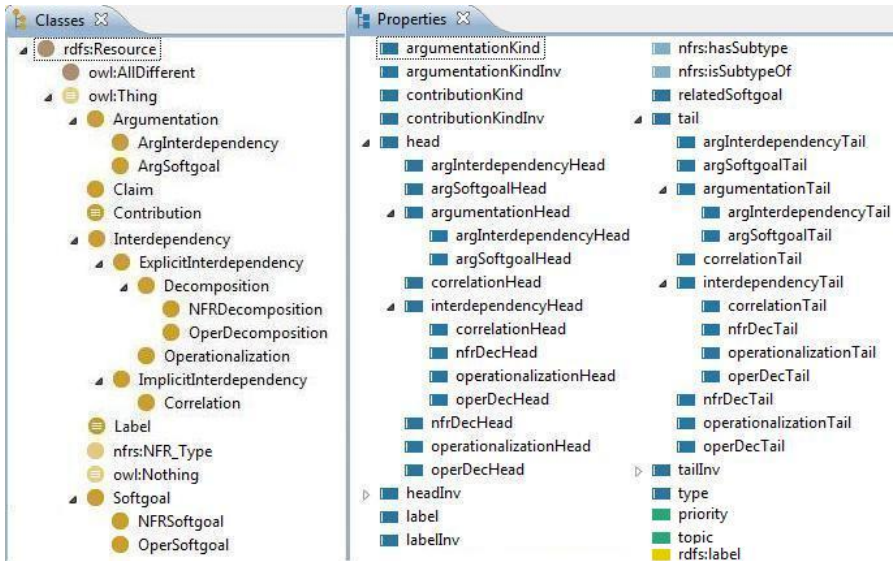


Fig. 2. NDR Ontology: Classes and Properties

of the simple concepts. Thus, facets enable assigning one object to a multiple classifications. Prieto-Diaz employed faceted classification schema to organize a software reuse library [15]. Conclusions of this work stated faceted classification contributed to improve search and retrieval capabilities of the software library.

We have defined a bigger and more flexible set of facets than the one presented in [4]. The facets and values are:

1. **SIGs** are classified according to three sub-facets:
 - **Domain** sorts SIGs by their related domain.
 - **Information Kind** lists SIGs according to the existence of evaluation or argumentation rationale.
 - **Author** classifies SIG depending on the author/source of the SIGs.
2. **Softgoals** is organized in four sub-facets:
 - **NFR Type** lists Softgoals according to their related NFR type.
 - **Softgoal Type** sorts Softgoals depending on their type: NFR or Operationalizing Softgoal.
 - **Topic** shows Softgoals ordered by their related topic.
 - **Label** classifies Softgoals according to their evaluation label: Satisfied (with sub-values Satisfied, Weakly Satisfied) Undecided, Denied (with sub-values Weakly Denied, Denied) or Conflict.
3. **Interdependencies** sorts Interdependencies depending on their type: Correlation, Decomposition and Operationalization.
4. **Argumentation** lists Argumentation according to their type.

5. **Contribution** classifies Interdependencies and Argumentation by their Contribution: Negative (with sub-values Break, Hurt, Some) Unknown, Positive (with sub-values Help, Make, Some+, Or, And, or Equal).

Facets and their values are generated by using SPARQL [16] queries. SPARQL is a query language that can be used to query required and optional graph patterns and constraint solutions to user-defined filters. For example, Interdependencies facet values are generated by:

```
SELECT DISTINCT ?interType
WHERE {{?interType rdfs:subClassOf ndr:ExplicitInterdependency.}
UNION {?interType rdfs:subClassOf ndr:ImplicitInterdependency.}}
```

Multi-facet search can be generated by combining multiple facet values. If the architect searches Correlations that point to Accuracy NFR Softgoal and contribute with a Help label, he can choose the facets values Correlation for Interdependency facet, Accuracy and NFR Softgoal for Softgoal facet, and Help for Correlation facet. The SPARQL query that implements this multi-facet search would be:

```
SELECT DISTINCT ?interType ?softgoal ?softgoalLabel
WHERE {?interType rdf:type ndr:Correlation.
?interType ndr:correlationHead ?softgoal.
?softgoal rdf:type ndr:NFRSoftgoal.
?softgoal ndr:type nfrs:NFRAccuracy".
?interType ndr:contributionKind ndr:Help.}
```

This capability enables architects to implement different classifications depending on their concerns. The architect can choose values in different facets and one or more values per facet. This ontology-based generation of multi-facet classification allows to quickly create new facets and values by using concepts and relationships of the NDR ontology.

4.3 Ontology-Based Fragment Identification

NFR and design rationale knowledge usually become to a huge amount of information. Recovering valuable information inside NFR models can be hard. We have defined NFR knowledge fragments templates to represent relevant knowledge chunks to be recovered (and reused) by architects. The proposed valuable NFR knowledge fragments are:

1. **SIG** shows the entire SIG.
2. **Softgoal** displays a Softgoal, its related Argumentation and its properties such as label, topic and the related NFR_type.
3. **Interdependency** shows an Interdependency and its Contribution, its related Argumentation, the Softgoal to which it points and the Softgoal from which it sets off.
4. **Argumentation** displays an Argumentation and its contribution, the Claim and the related Refinement or Softgoal.

5. **Trade-Off** shows a NFR Softgoal that is or could be labeled as Conflict², its Refinements, Correlations, Argumentations and properties.

Fragment templates are defined as subgraph patterns of the semantic graph that represents the SIG model. The subgraph is recovered by using a SPARQL query. For example, the query to generate the trade-off fragment template is shown below. It describes all properties of the softgoal with label Conflict and the properties of its interdependencies.

```
DESCRIBE ?softgoal ?interdependency ?softgoaltail
WHERE {?softgoal rdfs:type "Accuracy".
?softgoal ndr:label ddr:Conflict.
?interdependency ndr:head ?softgoal.
?interdependency ndr:tail ?softgoaltail.}
```

5 Prototype

A prototype implementation of the ontology-based faceted classification and fragment identification has been developed. This prototype can support the scenarios described in Section 2. For example, if the architect requires searching

Fig. 3. Faceted Search - Prototype

² Designers can deal with a trade-off and change a Conflict into a Satisfied or Denied Label. Inference rules can identify this situation and treat it as a conflict resolution.

conflicts related to Cost, he can create a multi-facet search. He can choose Accuracy, NFR Softgoal and Conflict for the facet Softgoal and search for Trade-offs. The tool will execute the next query to recover the softgoals related to the NFR Accuracy with conflicts:

```
SELECT DISTINCT ?softgoal ?softgoalLabel
WHERE ?softgoal rdf:type ndr:NFRSoftgoal.
?softgoal ndr:type nfrs:NFR_Accuracy.
?softgoal ndr:label ndr:Conflict
```

The user can choose one of the results. Once he chooses a specific trade-off, the tool will execute the query to describe the NFR knowledge fragment. In this case, the tool would run the query shown in Subsection 4.3. Figure 3 shows this multi-faceted search and the trade-off fragment for the SIG shown in 1.

6 Conclusions

This article has proposed an ontology-based faceted classification and fragment identification of NFR knowledge. They allow a configurable search and identification of relevant NFR knowledge. SIG models are translated to instances of the NDR ontology. Thus, the NFR and design rationale knowledge is stored as machine-readable semantic graphs. Facets and their values as well as NFR knowledge fragment templates are generated by using SPARQL queries to recover the required NDR ontology instances.

Machine-readability allows to current, and even future, applications to use the stored knowledge to facilitate complex data processing for humans. Ongoing work will add more functionality to this tool to turn it into a fuller semantic NFR knowledge store with improved human and machine-readability. This will enable intuitive search and exploration of NFR knowledge, analysis of alternatives, and ultimately enabling architects to collect, share and reuse NFR and design rationale.

NFR and design rationale visualization can be significantly improved by using ontology-based description. An user-defined faceted browsing would allow users to define their own multi-facet classifications and valuable knowledge fragments.

A challenging problem is finding out the designer's decisions that explain changes among several successive versions of a rationale model such as conflict resolution. Inference rules are being defined to augment the NDR ontology reasoning power and discover these designer's decisions. It would allow tracing the model evolution and add more reusable knowledge to NFR stores.

References

1. Bass, L., Clements, P., Kazman, R.: *Software Architecture in Practice*, 2nd edn. Addison-Wesley Professional, Reading (2003)
2. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: *Non-Functional Requirements in Software Engineering*. Springer, Heidelberg (1999)

3. Cysneiros, L.M., do Prado Leite, J.C.S., de Melo Sabat Neto, J.: A framework for integrating non-functional requirements into conceptual models. *Requirements Engineering* 6(2), 97–115 (2001)
4. Cysneiros, L.M., Yu, E., Leite: Cataloguing non-functional requirements as soft-goals networks. In: *Proceedings of the Workshop on Requirements Engineering for Adaptable Architectures at the 11th IEEE International Requirements Engineering Conference*, pp. 13–20 (2003)
5. Simon, H.A.: *The Sciences of the Artificial*, 3rd edn. The MIT Press, Cambridge (1996)
6. Balushi, T.H.A., Sampaio, P.R.F., Dabhi, D., Loucopoulos, P.: ElicitO: A quality ontology-guided NFR elicitation tool. In: *Requirements Engineering: Foundation for Software Quality*, pp. 306–319. Springer, Heidelberg (2007)
7. Dobson, G., Hall, S., Kotonya, G.: A domain-independent ontology for non-functional requirements. In: *ICEBE 2007. Proceedings of the IEEE International Conference on e-Business Engineering*, pp. 563–566. IEEE Computer Society, Washington (2007)
8. Kruchten, P.: An ontology of architectural design decisions in software intensive systems. In: *Second Groningen Workshop on Software Variability*, pp. 54–61 (2004)
9. Kruchten, P., Lago, P., van Vliet, H.: Building up and exploiting architectural knowledge. In: Reussner, R., Mayer, J., Stafford, J.A., Overhage, S., Becker, S., Schroeder, P.J. (eds.) *QoSA 2005 and SOQUA 2005*. LNCS, vol. 3712, pp. 43–58. Springer, Heidelberg (2005)
10. Akerman, A., Tyree, J.: Using ontology to support development of software architectures. *IBM Syst. J.* 45(4), 813–825 (2006)
11. Sancho, P.P., Juiz, C., Puigjaner, R., Chung, L., Subramanian, N.: An approach to ontology-aided performance engineering through NFR framework. In: *WOSP 2007. Proceedings of the 6th international workshop on Software and performance*, pp. 125–128. ACM, New York (2007)
12. McGuinness, D.L., van Harmelen, F.: *OWL web ontology language overview*, W3C recommendation (February 2004), <http://www.w3.org/TR/owl-features/>
13. López, C., Cysneiros, L.M., Astudillo, H.: NDR ontology: Sharing and reusing nfr and design rationale knowledge. In: *MARK 2008. First International Workshop on Managing Requirements Knowledge* (2008)
14. Ranganathan, S.: *Prolegomena to Library Classification*. Asian Publishing House, Bombay (1967)
15. Prieto-Díaz, R.: Implementing faceted classification for software reuse. *Commun. ACM* 34(5), 88–97 (1991)
16. Prud'hommeaux, E., Seaborne, A.: *SPARQL query language for RDF*, W3C recommendation (January 2008), <http://www.w3.org/TR/rdf-sparql-query/>

Aspect-Oriented Modeling of Quality Attributes^{*}

Mónica Pinto and Lidia Fuentes

Dept. Lenguajes y Ciencias de la Computación, University of Málaga, Spain

[pinto,lff}@lcc.uma.es](mailto:{pinto,lff}@lcc.uma.es)

<http://caosd.lcc.uma.es>

Abstract. Quality attributes are usually complex enough to be decomposed in a set of related concerns, with complex interactions and dependencies among them. Moreover, some of these concerns have a crosscutting nature being tangled and/or scattered with other concerns of the application. Aspect-Oriented Software Development is a good option to model a software architecture that requires the identification, modeling and composition of crosscutting concerns. This paper defines a process for the aspect-oriented modeling of quality attributes, specially those that have high functional implications. The main goal is to produce "built-in" reusable and parameterizable architectural solutions for each attribute.

1 Introduction

The quality attributes (QA) of a software system must be well understood and articulated early in the software development process, so the architect can design an accurate architecture that satisfies them [1]. However, modeling a QA is not a straightforward task. They are usually complex enough to be decomposed in a set of concerns, with dependencies and interactions among them. This is specially true for some QA with major implications on the core functionality of applications. Dealing with these QA as non-functional concerns does not provide enough information about what kind of architectural artifacts have to be used to satisfy them [2]. Thus, several proposals suggest a complementary approach in which QA with strong implications on the application core are incorporated to the architecture as functional concerns [3,4,5,6]. However, we identify that these approaches present two important shortcomings.

Regarding the first shortcoming, existing solutions do not usually take into account the crosscutting nature of the functional concerns of a QA, which makes them to be tangled – several concerns encapsulated in the same software artifact, or scattered – same concern split across different software artifacts, within the core functionality of the system. This shortcoming can be solved using Aspect-Oriented Software Development (AOSD), which focuses on identifying, modeling and composing crosscutting concerns across the software life cycle.

However, modeling a QA using AOSD is more intricate than just adding an aspect to our architecture. Not all the concerns behave as *aspects*, some

^{*} Supported by AOSD-Europe project IST-2-004349 and AMPLE Project IST-033710.

of them are better modeled as components. Some concerns will be completely reusable in any context, others may require some level of parametrization based on information that depends on the application core model. Summarizing, there are many issues that need to be considered to model a QA in general, which also need to be considered in an AO architectural solution. Thus, it can be useful the definition of a software process for the AO modeling of QA. Similarly to [4] we propose a process for identifying and specifying built-in architectural patterns, particular to AO architectural approaches.

The second shortcoming is the representation of the proposed patterns for QA. They are mainly specified by filling a table with information about the architectural implications [6], or by textual descriptions of intricate scenarios [7]. Thus, a ready-to-use solution that the software architect can (re)use in different applications is not provided. In order to cope with this limitation we propose the use an AO architectural language with support to store the specified models in a repository for later instantiation and reuse. Concretely, we propose the use of the AO-ADL [8] language and the AO-ADL Tool Suite.

After this introduction, the second section motivates our approach. The conceptual framework and its use to model the usability QA are discussed in section 3 and section 4 respectively. Section 5 describes how to instantiate this process using AO-ADL and the AO-ADL Tool Suite. Finally, in section 6 we evaluate our proposal and in section 7 we present our conclusions and future work.

2 Motivation and Related Work

The importance of modeling the functional part of QA at the architectural level has been identified in both the non-AO and the AO communities. An example in the non-AO community is [3,4] where all the functional implications of the usability attribute have been studied and documented. Another example is the reports generated from QA workshops [9], where a taxonomy of concerns and factors that influence the satisfaction of those concerns, have been defined for security, safety, dependability and performance. Examples in the AO community are the studies of crosscutting concerns in [7,10], which include security, persistence, context awareness and mobility.

To motivate our approach we have studied several QA with functional implications that are complex enough as to be decomposed in a set of related concerns. In Figure 1 the circles in dark grey are examples of QA and the circles in white are concerns identified for each attribute. The slashed lines represent dependencies between attributes. We want to highlight that:

1. The functional nature of the concerns in Figure 1 suggests that for many QA it is not possible to reason about them at the architectural level without adding attribute specific functionalities to the core application.
2. The same concerns are shared by several QA and their modeling is usually repeated for each framework. For instance, fault-tolerance is modeled as a concern of the usability attribute in [3], of the security attribute in [7], and

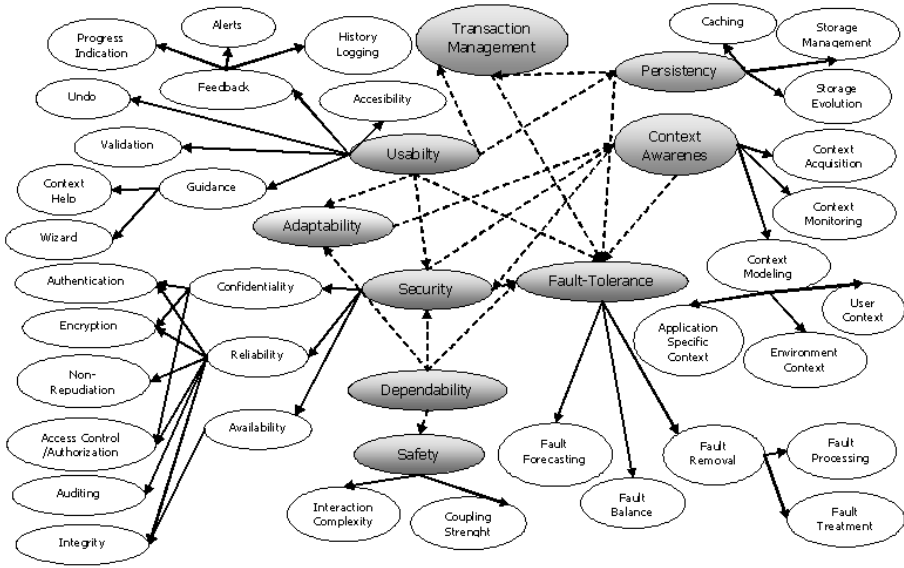


Fig. 1. Concern and Dependencies Quality Attribute Graph

of the context awareness attribute in [7]. One of the reason for this is the lack of a repository of (re)usable solutions.

3. The constraints that a QA imposes in the core application and the impossibility of defining completely reusable architectural solutions for some concerns are not always appropriately raised.
4. The format for presenting architectural solutions does not help to understand the peculiarities of each concern. The details about different alternatives to satisfy a particular concern, about the dependencies with other concerns and with the core application are not well-documented from early stages.

3 A Process for AO Modeling of Quality Attributes

Figure 2 illustrates the main activities defined in our software process. As previously discussed, complex quality attributes are usually decomposed in a set of domain specific related concerns. So, the architect should avoid the tendency of modeling the QA by a single black-box aspect. Thus, the first step in the process is the selection of a taxonomy of concerns, either an existing one defined by experts [9], or a new one identifying the relevant concerns and the dependencies and interactions among them (activity 'Reuse/Define Taxonomy of Concerns').

Based on the information provided by the taxonomy of concerns, the next step is deciding if the QA is suitable or not to be modeled following our approach. This will depend on whether or not the attribute have important functional implications, and on the number of concerns that have a crosscutting nature [1]. Also, it

¹ Notice that the criteria to answer these questions is out of the scope of this paper.

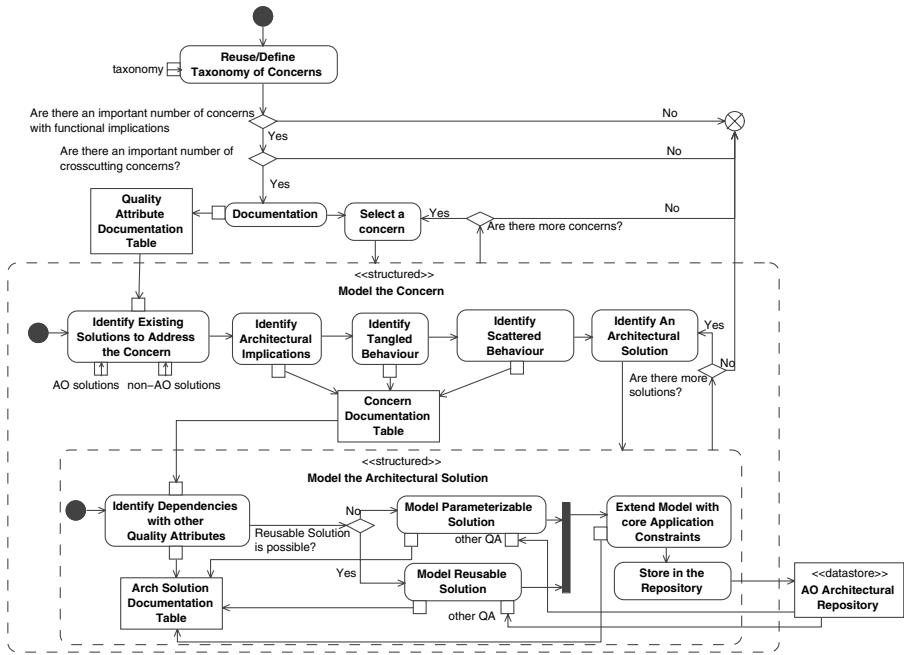


Fig. 2. Activity Diagram of the AO Modeling Process

is important to document the previous steps (activity 'Documentation' with output 'Quality Attribute Documentation Table'). The documentation consists on either a link to a taxonomy of concerns, or in case the taxonomy was defined from scratch, this is the correct place to include a complete description of it.

At this point is time to start the modeling of the main concerns already identified. The structured activity with name 'Model the Concern' in Figure 2 describes how to model each concern. The idea is once again to avoid modeling a concern from scratch. Thus, the first step to model a particular concern is to study existing AO and non-AO solutions (activity 'Identify Existing Solutions to Address the Concern' with inputs 'AO solutions' and 'non-AO solutions'). These solutions do not necessarily need to be available at the architectural level, but at design or implementation level. Thus, we need to identify the architectural implications of the concern in order to identify the main functionalities to be modeled at the architectural level (activity 'Identify Architectural Implications'). After this, since we are using an AO approach, the next step is to identify the tangled (activity 'Identify Tangled Behavior') and scattered (activity 'Identify Scattered Behavior') behaviors. All this information is the input to model a particular architectural solution. Once again, all this information is documented in the table we have defined for this purpose (object 'Concern Documentation Table'). In this case, links to existing solutions are provided and the architectural implications, the tangled and scattered behaviors of this solution are described. This will help to understand our solutions, as well as to compare them with other existing proposals.

Depending on the concern, more than one architectural solution may be considered. For each of them, the 'Model the Architectural Solution' structured activity describes how to model them. In addition to all the information already collected, for each particular architectural solution, the dependencies with concerns in other QA need to be identified and conveniently modeled to avoid repeating the same concerns in different quality models (activity 'Identify Dependencies with other quality attributes'). This is a very important step that it is not always considered in other approaches, with the serious problem that concerns that appear in several QA are modeled several times from scratch.

With this information the software architect is ready to decide whether a reusable or a parameterizable solution should be modeled. In our process we define a *Reusable Solution* as one in which the concerns can be modeled completely independent of the behaviors of the core functionality. We mean that the components and their interfaces can be incorporated into the system architecture without additional updates. This means that no parametrization is required. There are however other concerns whose behavior needs to be adapted to each particular application. In these cases the sub-architecture modeling the concern has to be seen as a template or pattern that have to be instantiated before using it in a particular application architecture. We name this a *Parameterizable Solution*. Following the same simile than before, these are concerns whose instantiations will differ depending on the part of the core functionality where are applied. Of course, the AO architectural language used to specify these solutions must support the definition of parameterizable architectures.

Either if the concerns can be directly reused or need to be parameterized, the addition of some of them to an existing architecture may require that this architecture satisfies certain constraints (e.g. expose its state for the persistence concern). These constraints should also be modeled and documented (activity 'Extend Model with Core Application Constraints'). This information will be used when the QA sub-architecture is bound to the core application architecture to determine if the later should be extended or not to satisfy those constraints.

Finally, the modeled QA is obtained, and then stored in a repository ('AO Architectural Repository') of reusable architectural solutions (activity 'Store in the Repository'). This is an important contribution of our approach that considerably increases the possibilities of reusing the QA models in different contexts, since all the dependencies either with the core application or with other QA' concerns are considered. Other main difference with other non-aspect and AO architectural solutions, is that models are available to be directly reused.

4 Modeling the Usability Attribute

In this section we follow the process defined in Figure 2 to model the usability QA. The input to the process is the usability taxonomy defined in the framework 4 that includes concerns such as feedback, error management, availability and guidance, among others. These are concerns with important functional implications and that clearly crosscut the functionality of the core application, which will help us to illustrate the issues discussed in previous section.

Firstly, in Figure 3 we model the cancelation functionality. According to 4, providing users with support to cancel their actions helps to satisfy the 'user control' usability property. This functionality is tangled with all the components that contain actions that may be canceled by the user. Moreover, it is scattered between the model and the view, since the model specifies the actions to be canceled and the view needs to show the list of actions that may be canceled.

According to our process, this is an example of a reusable solution. Modeling the cancelation functionality is independent of the specific functionality in the core application (e.g. Shopping Cart application). Concretely, is modeled by two aspectual components (Cancelation Aspect and Cancelation View) and one component (State Recording). The rest of components in Figure 3 model the constraints imposed to the core application by this solution, and are discussed in the next paragraph. In Figure 3, the Cancelation Aspect is an aspectual component that advises the actions on the core model that may be canceled by the users, according to the list of advices defined in the `CancelInt` interface. Thus, before an action is initiated in a component of the core application, the aspectual component is in charge of recording the component state by intercepting the beginning of the action and by interacting with the `State Recording` component. Also, when an action finishes and the component state is not required anymore, the aspectual component is in charge of invalidating the component state by intercepting the end of the action and by interacting with the `State Recording` component. This architectural solution also needs to consider that an action in one component may require interactions with other components, and that canceling that action may also require to update the state of the other components. Thus, the aspectual component also intercepts the interaction between components, but only when they occur inside an action that is considered as cancelable. For that, the type of advice to use is `cflow` [11] that captures join points but only when they occur in the control flow of another particular join point. Moreover, this aspectual component interacts with the `Cancelation View` aspectual component to update the user view with the new cancelable action. The cancelation is initiated by the user by interacting with this component through the `cancel()` operation of the `CancelViewInt` interface. The cancelation functionality is injected to the core application as an aspect introduction.

We also use the example in Figure 3 to illustrate how the constraints imposed by this concern in the core functionality are modeled as part of the cancelation functionality. Concretely, canceling an action it is not possible if components in the core application do not expose the state affected by the actions that can be canceled. This is specified in Figure 3 by modeling the `<StateInt>` interface provided by the components in the core application. The use of `<>` in the name of the interface indicates that it is a parameter that need to be instantiated when the architectural solution is used in a particular context. `<Interface>` is another parameter modeling the interface that contains the cancelable actions. We have included the parameters `<Component1>` and `<Component2>` to model the interactions between components captured by the `cflow` advice.

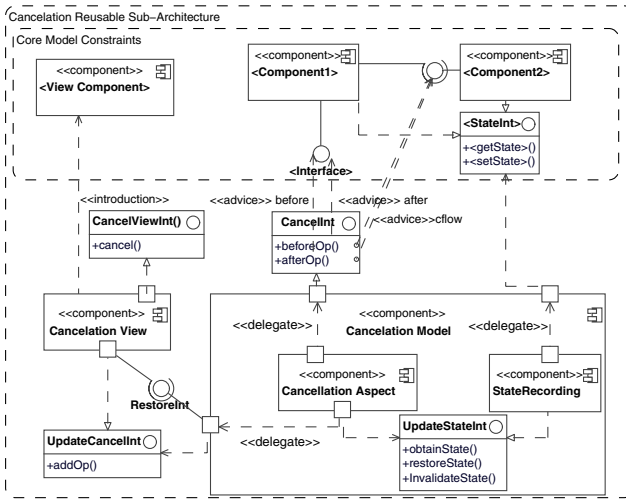


Fig. 3. Example of Reusable Architecture with constraints on the core model

In Figure 4 we show two examples of parameterizable sub-architectures by modeling the data validation and the data formatting concerns associated with usability in 4. The architecture modeling these concerns can not be completely reused in different contexts because their behavior heavily depends on the particular application data they intercept to validate or format. Thus, we have parameterized the definition of the components, interfaces and operations of the architecture by using <> as part of the names. These parameters will then be instantiated with different validation/format components depending on the different parts of a same application or on the different applications that require validation/format, as shown in section 5. UML 2.0 templates may have been used but, in order to create the repository of architectural solutions, in this paper we define and instantiate the architectural templates using AO-ADL. Figure 4 also shows the information that is recorded in the Concern Documentation Table (4.1 to 4.4) and the Arch Solution Documentation Table (4.4.1 to 4.4.4) tables as defined in the process defined in Figure 2. We omit the Quality Attribute Documentation Table (1 to 4) due to lack of space. These tables provide a systematic way of documenting all the information used and generated by the process.

Finally, we discuss the dependencies identified between usability and other QA. Based on the framework in 4, we have identified that usability depends at least on the adaptability and the fault tolerance QA. Moreover it has an indirect dependency with security and context-awareness. It depends on the adaptability attribute because one of the usability concerns is to be able to adapt the application to the user preferences and the user context. It depends on security because it is not possible to adapt the application to the user preferences if users are not previously authenticated. It depends on context-awareness because information about the user context is needed to adapt the application. Finally, it depends on the fault tolerance attribute because error management, prevention

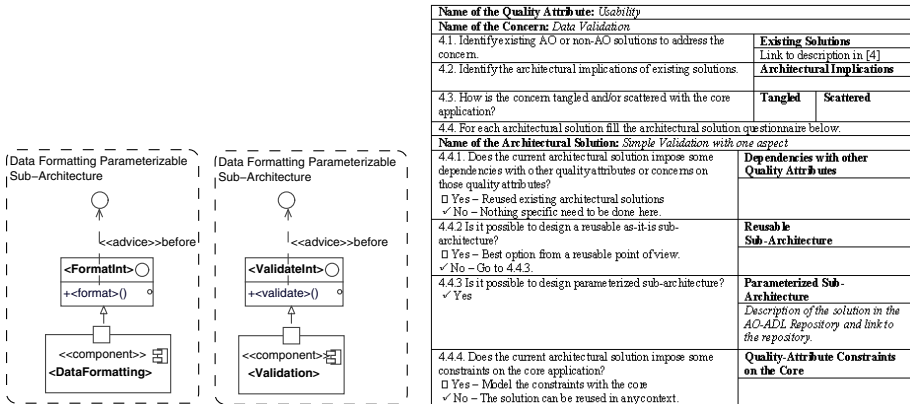


Fig. 4. Examples of Parameterizable Sub-Architectures and Documentation Table

and correction are concerns identified as relevant for usability in [4]. According to the process in Figure 2, during the modeling of the usability attribute the AO security, context awareness and fault tolerance models would be reused by importing and instantiated them from the AO Repository of Architectural Solutions. If those models would not exist in our repository we have to follow the same process to model them. This approach differs from the approach followed in [4] where these concerns are modeled from scratch, without considering existing taxonomies or solutions for other related concerns.

5 Process Support Using AO-ADL

In order to support the software process described in previous section, the following requirements must be fulfilled: (1) An AO architectural language able to specify the aspectual bindings between the QA sub-architectures and the core application architecture; (2) An architectural modeling language able to define and instantiate architectural templates, and (3) Tool Support to store templates and instantiated sub-architectures for later (re)use in different contexts.

AO-ADL [8] is an architecture description language that satisfy these requirements. AO-ADL defines a symmetric model where both non-crosscutting and crosscutting concerns are modeled by components. Therefore, instead of inventing a new structural element to model aspects, AO-ADL defines a new relationship for weaving aspectual components. Concretely, 'crosscutting' becomes a new kind of relationship between components that needs to be modeled at the architectural level. It is modeled by extended traditional connectors with a new kind of role, the aspectual role. The most important contribution of AO-ADL is the components and connector templates. Another important contribution of AO-ADL is the implementation of a repository of components, connectors and templates that can be imported and reused in the specification of different software architectures. The

AO-ADL language is supported by the AO-ADL Tool Suite, implemented as an Eclipse plug-in² that can provide a catalogue of architectural templates that are ready to be instantiated.

6 Discussion and Evaluation

We evaluate our proposal applying the usability AO patterns to existing AO-ADL software architectures of a Toll Gate System (TG) [12], a Health-Watcher Application (HW) [13] and an Auction System (AS) [14]. We focus on if the core architecture: (1) already provided some functionality related to usability or not; (2) satisfied all the constraints imposed by the usability architectural patterns, and (3) modeled some crosscutting concerns from scratch (Table 1).

Table 1. Incorporating usability to existing software architectures

	Provides Usability	Satisfy All Constraints	Attributes From Scratch
TG	Feedback, Error Handling	No (Cancellation)	Usability, Security, Fault Tolerance, Persistence
HW	Consistency, Data Formatting, Error Handling	Yes	Usability, Security, Fault Tolerance, Persistence
AS	No	Yes	Security, Fault Tolerance, Persistence

In the first column of Table 1 we can observe that the TG and the HW already included some functionality related to usability (feedback and error handling in TG and consistency, data formatting and error handling in HW). Contrarily, usability concerns were not present in the software architecture of the AS. In this case, no particular issues were found when (re)using the AO architectural solutions to incorporate usability to the AS. However, in order to reuse the usability architectural solutions in the TG and the HW, some level of modification in the core architecture was required, in order to omit the components modeling the usability concerns from the core. The ideal situation would have been to identify, during the requirement stage, that these concerns helped to satisfy a QA, importing them from our repository instead of defining them from scratch. For instance, the consistency and data formatting concerns in the requirements of the HW would be instantiations of the architectural solutions in Section 4.

The information in the second column is the result of trying to incorporate usability concerns that impose some constraints on the core application. Some limitations were found adding 'cancellation' to the TG, because the components in the existing architecture did not expose their state and this was a requirement of our cancellation architectural pattern. Fortunately, this constraint was well-documented in our approach being able to easily identify that the original software architecture need to be extended with the components' state.

² Visit our Eclipse Update Site in <http://caosd.lcc.uma.es/AO-ADLUpdates>

Finally, the third column shows that not only the usability attribute was identified in these software architectures. Other QA or crosscutting concerns such as security, fault tolerance and persistence were also identified and can take advantage of the software process defined in this paper.

7 Conclusions and Future Work

In this paper we have illustrated the complexity of modeling a QA, and have defined a process to guide software architects using AOSD to model QA with crosscutting nature and important functional implications. We propose using the AO-ADL Tool Suite as the repository of reusable and parameterizable architectural solutions, though other AO architectural approaches satisfying the requirements imposed by our process may be used. As future work we plan to apply our process to other quality attributes in order to refine the process, as well as in order to extend the repository of reusable patterns with new concerns.

References

1. Bachmann, F., et al.: Designing software architectures to achieve quality attribute requirements. *IEE Proceedings* 152(4), 153–165 (2005)
2. Cysneiros, L.M., Werneck, V.M., Kushniruk, A.: Reusable knowledge for satisficing usability requirements. In: *RE 2005* (2005)
3. Juristo, N., Moreno, A.M., Sanchez, M.I.: Guidelines for eliciting usability functionalities. *IEEE Transactions on Software Engineering* 33(11), 744–757 (2007)
4. Juristo, N., Lopez, M., Moreno, A.M., Sanchez, M.I.: Improving software usability through architectural patterns. In: *ICSE Workshop on SE-HCI*, pp. 12–19 (2003)
5. Welie, M.V.: *The amsterdam collection of patterns in user interface design* (2007)
6. Folmer, E., Bosch, J.: Architecting for usability; a survey. *Journal of Systems and Software* 70(1), 61–78 (2004)
7. Geebelen, K., et al.: Design of frameworks for aspects addressing 2 additional key concerns. *Technical Report AOSD-Europe D117* (February 2008)
8. Pinto, M., Fuentes, L.: AO-ADL: An ADL for describing aspect-oriented architectures. In: *Early Aspect Workshop at AOSD 2007* (2007)
9. Barbacci, M., et al.: Quality attributes. *Technical Report CMU/SEI-95-TR-021* (December 1995)
10. Tanter, E., Gybels, K., Denker, M., Bergel, A.: Context-aware aspects. In: Löwe, W., Südholt, M. (eds.) *SC 2006*. LNCS, vol. 4089, pp. 227–242. Springer, Heidelberg (2006)
11. Kiczales, G., et al.: An overview of AspectJ. In: Knudsen, J.L. (ed.) *ECOOP 2001*. LNCS, vol. 2072, pp. 327–355. Springer, Heidelberg (2001)
12. Pinto, M., et al.: Report on case study results. *Technical Report AOSD-Europe D118* (February 2008)
13. Pinto, M., Gámez, N., Fuentes, L.: Towards the architectural definition of the health watcher system with AO-ADL. In: *Early Aspect Workshop at ICSE* (2007)
14. Chitchyan, R., et al.: Mapping and refinement of requirements level aspects. *Technical Report AOSD-Europe D63* (November 2006)

Improving Security of Oil Pipeline SCADA Systems Using Service-Oriented Architectures

Nary Subramanian

Department of Computer Science
The University of Texas at Tyler
3900 University Blvd.
Tyler, Texas 75799, USA
nsubramanian@uttyler.edu

Abstract. Oil pipeline Supervisory Control and Data Acquisition (SCADA) systems monitor and help control pipes transporting both crude and refined petroleum products. Typical SCADA system architectures focus on centralized data collection and control – however, this system has vulnerabilities that decrease the overall security of the system, especially for an oil pipeline SCADA. Service-oriented architecture (SOA) helps to improve security of SCADA systems by providing more localized data collection and control. In this paper we describe an SOA-based architecture for oil pipeline SCADA system that provides improved security compared to traditional architectures. An SOA-based SCADA divides the entire length of the pipeline system into zones where services offered within a zone are controlled by the zone master and masters periodically batch-update the central database over the back-bone network. The feasibility is explored by mathematical analysis and emulation.

Keywords: SCADA, petroleum, pipeline, architecture, services, security.

1 Introduction

Crude oil is terrestrially distributed by pipelines: from drilling rigs to crude oil storage tanks, from storage tanks to refineries, and finally the refined oil from refineries to gasoline storage tanks. Typically these pipelines span several thousands of miles – the US alone has about 150,000 miles of pipelines for transporting petroleum products [1]. In order to efficiently monitor and control this huge oil pipeline network supervisory control and data acquisition (SCADA) systems are employed. The oil pipeline SCADA has several hundred RTU's (remote terminal units) [14] that are connected to field instruments that measure pressure, temperature, and rate of flow of the oil flowing through the pipes, as well as change the statuses of valves and pumps along the pipeline. The RTU's communicate with a central master station using communication links such as satellite, cable, cellular, or fiber optic transmission media. The system architecture for traditional SCADA system is shown in Figure 1. A typical installation has several hundred RTU's communicating over dedicated links to a central master station [10, 11]. The most important aspect of oil pipeline is security [2, 3, 4, 5, 6, 7] and therefore SCADA systems are designed to provide real-time security status of the entire pipeline so that necessary action may be taken by the human agents monitoring the central information.

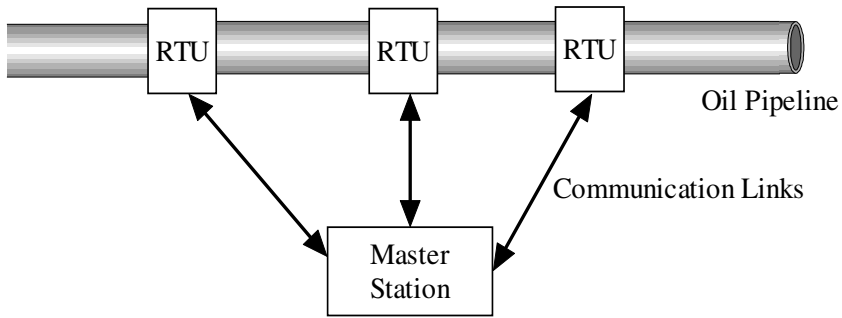


Fig. 1. Typical Oil Pipeline SCADA System Architecture

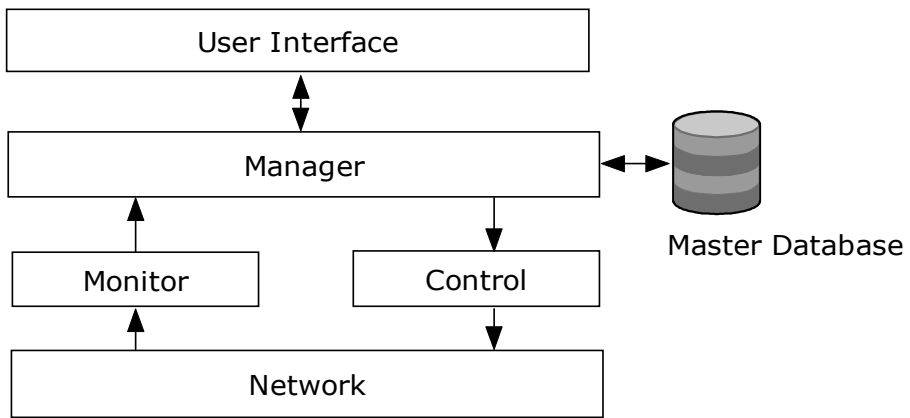


Fig. 2. Typical Software Architecture for SCADA System

Traditionally, software architecture of a SCADA system is a three layer architecture where the bottom layer is the data layer, the middle layer is the processing layer, and the top layer is the user interface layer. The layered software architecture for the SCADA system is shown in Figure 2. The processing layer accesses data from all RTU's regarding the status of various sensors and controls, and issues commands to the controls to change their states. The data received from the sensors and controls are stored by the processing layer in the data layer; besides, this data is also sent to the user interface layer for display to humans. Based on human responses to the data display, the user interface layer instructs the processing layer to change statuses of specific controls upon which the processing layer issues the appropriate commands to the relevant RTU's. This three layer architecture software resides in the master station of the SCADA system and all RTU's are assumed to be slaves in the system that send messages to and receive commands from the master. Therefore, the entire operation of the SCADA system is dependent on the network that connects the RTU's with the master. Oil pipeline SCADA systems communicate over several hundreds to thousands of miles and therefore need wide-area networking or the Internet to support their operations [10, 11]. Even though basic authentication mechanisms exist, security

in oil pipeline SCADA systems are almost exclusively related to network security and several recent security breaches [5, 8] have occurred through the network. Therefore, the following main techniques have been suggested to improve oil pipeline security:

1. reduce network traffic: as discussed in [6, 7, 8] network is perhaps the most important component in a modern SCADA system from a security viewpoint. Therefore reducing network traffic will help improve SCADA security.
2. include people along the pipeline route in the security strategy: one of the latest strategies to improve oil pipeline security is to include local communities along the oil pipeline for the latter's operation and maintenance [3]. By allowing local communities develop a sense of ownership in the pipeline, the security of the pipeline improves.
3. avoid centralized control so that there is no one vulnerable critical point: the master station tends to be one of the main facilities of oil pipeline SCADA that serves as a vulnerable critical point. As pointed out in [15] such vulnerabilities need to be removed to improve security of pipeline installations.

In this paper we propose a new software architecture for oil pipeline SCADA systems that employs the concept of services, divides the entire length of the system into zones, and several zones may be collected into groups. Software using service-oriented architecture (SOA) [12] now runs in the processing layer of each zone and each zone master controls only that zone: this significantly reduces long-distance network traffic. Moreover, each zone is controlled by people from that area and this encourages local people to take ownership in the operation and security of the pipeline. Periodically, zones may send information to their group master; however, there can now be any number of group masters and this avoids having one centralized master station; moreover, masters may be dynamically reconfigured. The major advantages are improved security, improved reliability by avoiding single point of failure and improved maintainability of the system by involving local businesses along the length of the pipeline. The feasibility of the SOA-based SCADA system for oil pipelines is explored using mathematical analysis and actual implementation.

This paper is organized as follows: Section 2 discusses SOA-based software architecture for oil pipeline SCADA, Section 3 discusses feasibility analysis of the SOA-based system, Section 4 presents our observations on the SOA-based system, and Section 5 presents our conclusions and possible directions for further research.

2 SOA-Based Software Architecture for Oil Pipeline SCADA

In the SOA-based system, the entire length of the pipeline is divided into several zones and there are several group masters. Each zone monitors and controls only its zone and therefore most of the data traffic is localized. Each group master monitors the status of all zones under its responsibility – the group master keeps track of the status of each zone under it: the status may be as simple as knowing the overall security of each zone or as complicated as completely replicating each zones' user interface in detail. Group master is not a separate workstation but one of the zones taking on the responsibility of being the group master. In the trivial case, there is only one

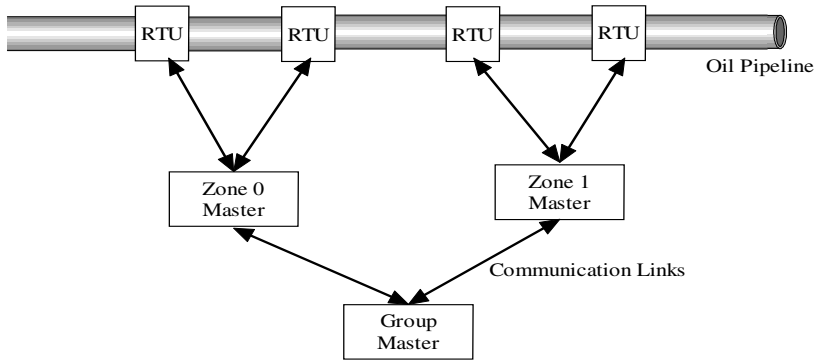


Fig. 3. System Architecture for Oil Pipeline SCADA Employing SOA

zone and that zone is also the group master – this corresponds to the traditional SCADA configuration. Each zone belongs to only one group master and each group master can have arbitrary number of zones assigned to it. The system architecture for SOA-based SCADA is shown in Figure 3.

The software architecture for the SOA-based system is shown in Figure 4. Each zone has its own user interface, manager, and database. In addition, each zone has its own web service broker and web services directory. All services for monitoring of RTU's, control of RTU's, and network access are registered in the web services directory and the web service broker accesses these services whenever needed. Likewise each zone registers its status interface with the group web services directory over the backbone network and the group master accesses the status of each of the zones assigned to it using these interfaces. The group master logs the details of its interactions with the zones on its database.

The SOA-based configuration significantly reduces communication requirements. The distances are now localized within each zone the size of which is set based on the needs of a specific system (a zone could be a critical portion of the system, a state, a province, a country, a geographic region, or the entire system itself), and the only inter-zone data transfer is that of the status of each zone. As described in the validation section of this paper, mathematically it can be shown that for normal inter-zonal data the communication requirements are reduced by about 75% for a four-zone system.

The SOA-based configuration actively encourages local businesses to participate in the oil pipeline security management by allowing them to register their services such as alternate network access or data analysis with the zonal web services directory that the zone manager can access if needed. As discussed in [12] only trusted businesses are allowed access to the web service broker for registration purposes and the security of the system will not be compromised by this procedure.

The SOA-based configuration also allows dynamic reconfiguration so that the role of the group master may be assigned to any of the zone masters. For this purpose each zone registers its interfaces with each other and this permits a zone master to access the status of each zone assigned to it. If a zone gets affected so that no communication is possible (or the zone must be shut-off for some reason), the group masters can quickly reassign zones between themselves for easier control of the entire system.

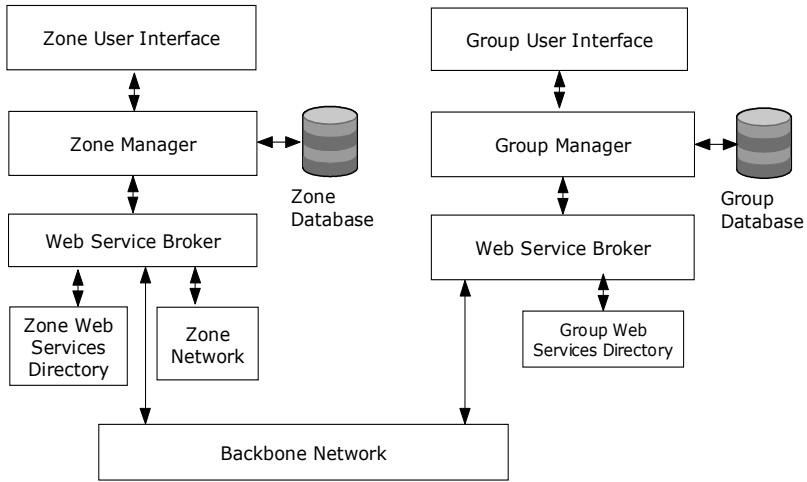


Fig. 4. Software Architecture for SOA-Based Oil Pipeline SCADA

3 Feasibility Analysis of SOA-Based SCADA System

Mathematical analysis of the communication requirements for a SOA-based oil-pipeline SCADA is given below. For the purposes of this analysis we assume that the oil-pipeline is 1000 miles long and there is an RTU for every mile (this assumption is validated by practice since RTU’s are usually uniformly distributed along the length of the pipeline [2]). We assume each RTU makes six measurements.

For traditional SCADA configuration, a central master, located 500 miles from either end of the pipeline,

$$\begin{aligned}
 \text{Total data communication requirement} &= 2 \times 6[1 + 2 + 3 + \dots + 500] \text{ reading-mile} \\
 &= 1,503,000 \text{ reading-mile} \\
 &= 12,024,000 \text{ bit-mile,}
 \end{aligned}$$

assuming one reading takes a byte.

For a four zone SOA-based configuration, with zones distributed uniformly, that is, each zone is responsible for 250 miles of the pipe length,

$$\begin{aligned}
 \text{Zone data communication requirement} &= 2 \times 6[1 + 2 + 3 + \dots + 125] \text{ reading-mile} \\
 &= 756,000 \text{ bit-mile}
 \end{aligned}$$

$$\text{Total zonal data communication requirement} = 4 \times 756,000 = 3,024,000 \text{ bit-mile.}$$

If each zone master updates a group master that is represented, for the purposes of this discussion, by a hypothetical master in the middle of the pipeline, then if each zonal update takes 1 byte, then

$$\begin{aligned}
 \text{Zone data update communication requirement} &= 2[375 \times 8 + 125 \times 8] \text{ bit-mile} \\
 &= 8000 \text{ bit-mile}
 \end{aligned}$$

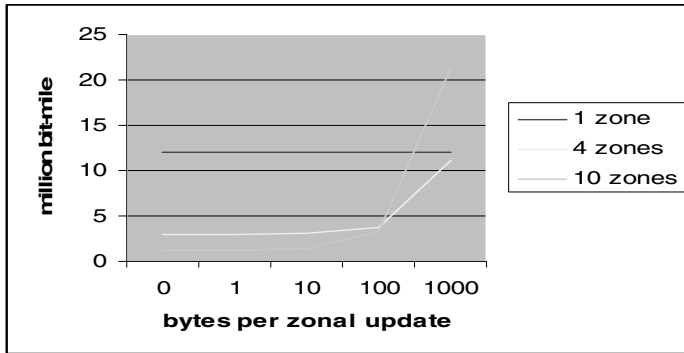


Fig. 5. Data Communication Requirements for Varying Zonal Update Sizes

Total data communication requirement for an SOA-based system that takes 1 byte to update the group master is: 3,024,000 bit-mile + 8000 bit-mile = 3,032,000 bit-mile.

Similar calculations were performed for 10 byte, 100 byte, and 1000 byte zonal updates for both 4 zones and 10 zones, and the results are shown in Figure 5.

As can be seen in Figure 5, 1 zone (equivalent to the traditional SCADA architecture) requires far more data communication requirements for normal zonal update data (< 1000 bytes): almost 75% more data communication requirement is needed by the traditional SCADA architecture.

We developed a physical implementation of an SOA-based system that emulated an oil-pipeline SCADA. This system transferred 1 byte per update and the group user interface is shown in Figure 6. As can be seen Figure 6 is sufficient for displaying the status of each of the four zones – zone that is not working efficiently is displayed in different color. This group interface conveys sufficient information to the human agent for monitoring at a high level each of the four zones.

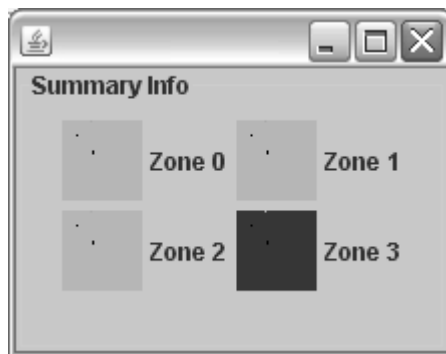


Fig. 6. Group User Interface Displaying Summary Zone Information

4 Observations

4.1 SOA-Based Architecture is Better than Distributed Architecture

The SOA-based SCADA for oil pipeline is a better alternative than a distributed architecture: in a distributed architecture the components are distributed but the interfaces are hard-coded. This not only does not help local businesses to easily provide their services but also does not help dynamic reconfiguration.

4.2 Improved Security

All the three requirements for improved security, namely, reduced communication requirements, participation of local communities, and decentralized master are satisfied by the SOA-based architecture. Communication requirements are only a fraction (25%) of the traditional SCADA architecture, local businesses can easily integrate their services with the architecture, and masters can be dynamically reconfigured.

4.3 Improved Reliability

Reliability of the system increases because of removal of bottlenecks – if one communication link fails, local businesses can be called upon to provide alternate communication means to the zone and group masters, and if one zone were to be blocked out, then communication with other zones is still possible. Since each zone takes responsibility for RTU's under its control, focus is more detailed and therefore, overall reliability improves.

4.4 Improved Maintenance

Maintenance is assigned to local businesses and therefore it is more timely and effective. Likewise, if any additional feature is required, local businesses can provide them as services that helps improve maintainability of the system.

4.5 Future Proofing

Recent technological advances such as VOIP (voice over IP) and instant messaging [9] are expected to boost ease of communication and ability to monitor and control SCADA systems. However, with SOA-based system, it is easy to incorporate the new and other future developments by simply providing an interface to these technologies as services. The services may even be outsourced to local businesses to hasten adoption.

The SOA-based software architecture may be adapted to any pipeline SCADA system such as water, natural gas, and sewage transmission systems. However, cost incurred in developing a distributed system using SOA needs to be weighed against potential benefits. In the case of oil pipelines, with the current cost per barrel of oil being in excess of \$140 [13], and with systems typically capable of pumping one million barrels per day [3] through oil pipelines, the cost of oil transmitted per day is in excess of \$140 million. For such valuable commodity, the cost of SOA-based system will be more than compensated by any potential losses due to security breaches.

5 Related Work

Oil conveying pipelines are part of critical infrastructure and SCADA systems are used extensively for monitoring and controlling the pipelines [3, 4]. Among the more important quality requirements of SCADA systems are security, reliability, and maintainability [16, 17, 19]. While several techniques have been proposed to improve security in SCADA systems – for example, redundancy [16] has been used to improve security for communication networks, intranet-based technology for real-time control to improve reliability and maintainability has been proposed in [18], and specific countermeasures for a set of vulnerabilities for electrical systems has been proposed in [19]. Detailed reliability analysis for SCADA systems have been performed in [20]. However, it has been suggested that SOA can be used in safety critical environments [21], and in this paper we considered SOA for improving security for oil pipeline SCADA systems – however, the use of SOA for SCADA necessitates a zone-based strategy so that geo-political interests are satisfied as well since pipelines frequently traverse national boundaries. The SOA-based approach offers promise to improve security, reliability, and maintainability of oil-pipeline infrastructure.

6 Conclusion

Supervisory control and data acquisition (SCADA) systems for oil pipelines monitor and control transfer of oil and petroleum products through the pipeline over several hundreds to thousands of miles. This is typically accomplished by having a central master station communicating over a variety of communication links with several hundred remote terminal units (RTU's) to monitor various physical parameters and to control valves and pumps along the pipeline to keep the oil flowing through the pipeline. Among the most important requirements to improve the pipeline security is to reduce network traffic, include local communities along pipeline route in the security strategy, and to avoid vulnerabilities such as having only a single master station. In this paper we propose an service-oriented architecture (SOA) based SCADA system for oil pipelines that helps to improve security, reliability, and maintenance. Our proposal includes a modified system architecture that divides the length of the pipeline into zones and groups where a group consists of several zones but one zone belongs to only one group. Each zone has a zone master and one of the several zone masters in a group also becomes the group master. By using services, the service brokers at the zone and group level and able to identify interfaces in other zones to form a dynamically reconfigurable architecture. The SOA based approach reduces network traffic, provides ability to local businesses to participate in the pipeline processes, and avoids vulnerabilities associated with having only one master. The feasibility of the SOA-based oil pipeline SCADA architecture was explored mathematically and by physical implementation. The network requirements for SOA-based system are typically only about 25% of the traditional SCADA architectures.

For the future we need to validate the system with more robust group master user interface that provides more details of the status of the zones and allows control of each zone as well. We also need to validate the dynamic reconfiguration of the group master by physical implementation and/or simulation. Another future activity is to

validate the SOA-based architecture when RTU's capture more readings as well as when RTU's are non-uniformly distributed. Also, the distinction between security and safety need to be delineated so that they are separately addressed. Moreover, it has been suggested that SOA is inefficient for security purposes [22] and that SOAP XML messages actually increase traffic size [23] – both these aspects need further investigation from an oil-pipeline SCADA standpoint. However, we believe that SOA-based SCADA is a promising and profitable option for improving oil-pipeline security.

Acknowledgements. The author wishes to thank the anonymous reviewers of the earlier version of this paper for their detailed and thorough comments.

References

1. http://en.wikipedia.org/wiki/List_of_countries_by_total_length_of_pipelines (accessed on July 5, 2008)
2. http://en.wikipedia.org/wiki/Pipeline_transport (accessed on July 5, 2008)
3. Ismailzade, F.: A Strategic Approach to Pipeline Security. Report of the Institute for the Analysis of Global Security (2004), <http://www.iags.org/n1115043.htm>
4. Clementson, D.P.: Reviewing SCADA basics. Pipeline and Gas Technology (2006) (accessed on July 5, 2008), <http://www.pipelineandgastechnology.com/story.php?storyfile=2defd4c7-bdad-4776-af94-fa948ad21b18.html>
5. Slay, J., Sitnikova, E.: Developing SCADA Systems Security Course within a Systems Engineering Program. In: Proceedings of the 12th Colloquium for Information Systems Security Education, pp. 101–108 (2008)
6. Idaho National Engineering and Environmental Laboratory. A Comparison of Oil and Gas Segment Cyber Security Standards. Report No. INEEL/EXT-04-02462 (2004)
7. API Standard 1164, Pipeline SCADA Security First Edition (September 2004) (accessed on July 5, 2008), <http://api-ep.api.org/filelibrary/1164PA.pdf>
8. Sauver, J.: SCADA Security (accessed on July 5, 2008), <http://darkwing.uoregon.edu/~joe/scada/>
9. Henrie, M.: API 1164 Standard Revision (accessed on July 5, 2008), http://www.api.org/meetings/topics/pipeline/upload/Morgan_Henrie_API_1164_Standard_Revision_API_Presentation_REv_1.pdf
10. Press Release, Sinopec selects Invensys for SCADA system on China's longest crude oil pipeline (October 11, 2005) (accessed on July 5, 2008), <http://news.thomasnet.com/companystory/468291>
11. References for Cegelec installations (accessed on July 5, 2008), <http://www.oilandgas.cegelec.com/References/ScadaRef.htm>
12. O'Neill, M., et al.: Web Services Security. McGraw-Hill, New York (2003)
13. Associated Press news report, Oil passes, settles above \$145 for first time (July 3, 2008) (accessed on July 5, 2008), <http://www.msnbc.msn.com/id/12400801/>
14. <http://en.wikipedia.org/wiki/SCADA> (accessed on July 5, 2008)
15. Matheson, M., Cooper, B.S.: Security Planning and Preparedness in the Oil Pipeline Industry. In: The Oil & Gas Review, pp. 104–108 (2004)

16. Farris, J.J., Nicol, D.M.: Evaluation of Secure Peer-to-Peer Overlay Routing for Survivable SCADA Systems. In: Proceedings of the 36th Conference on Winter Simulation, pp. 300–308. ACM Press, Washington (2004)
17. National Communications System, Technical Information Bulletin 04-1, Supervisory Control and Data Acquisition (SCADA) Systems (October 2004) (accessed on August 23, 2008),
http://www.ncs.gov/library/tech_bulletins/2004/tib_04-1.pdf
18. Ebata, Y., Hayashi, H., Hasegawa, Y., Komatsu, S., Suzuki, K.: Development of the Intranet-based SCADA (supervisory control and data acquisition system) for Power System. In: IEEE Power Engineering Society Winter Meeting, vol. 3, pp. 1656–1661. IEEE Press, Los Alamitos (2000)
19. Dagle, J.E., Widergren, S.E., Johnson, J.M.: Enhancing the Security of Supervisory Control and Data Acquisition Systems: the Lifeblood of Modern Energy Infrastructures. In: IEEE Power Engineering Society Winter Meeting, vol. 1, p. 635. IEEE Press, Los Alamitos (2002)
20. Bruce, A.G.: Reliability analysis of electric utility SCADA systems. IEEE Transactions on Power Systems 13(3), 844–849 (1998)
21. Prinz, J., Kampichler, W., Haindl, B.: Service Oriented Communication Architectures in Safety Critical Environments. In: Integrated Communications Navigation and Surveillance (ICNS) Conference (2006) (accessed on August 23, 2008),
http://spacecome.grc.nasa.gov/icnsconf/docs/2006/04_Session_A3/06-Kampichler.pdf
22. Roch, E.: SOA Security Architecture (2006) (accessed on August 23, 2008),
<http://it.toolbox.com/blogs/the-soa-blog/soa-security-architecture-11431>
23. Leonard, P.: High Performance SOA – A Contradiction in Terms? (2006) (accessed on August 23, 2008),
http://www.webservices.org/weblog/patrick_leonard/high_performance_soa_a_contradiction_in_terms

An Architecture to Integrate Automatic Observation Mechanisms for Collaboration Analysis in Groupware

Rafael Duque¹, María Luisa Rodríguez², María Visitación Hurtado²,
Manuel Noguera², and Crescencio Bravo¹

¹Departamento de Tecnologías y Sistemas de Información, Universidad de Castilla-La Mancha
E.S.I., Paseo de la Universidad 4, 13071 Ciudad Real, Spain
{Rafael.Duque, Crescencio.Bravo}@uclm.es

²Departamento de Lenguajes y Sistemas Informáticos, Universidad de Granada,
E.T.S.I.I.T., c/ Periodista Daniel Saucedo Aranda s/n, 18071 Granada, Spain
{mlra, mhurtado, mnuquera}@ugr.es

Abstract. The study of the effectiveness and suitability of the different facilities usually provided by groupware tools (e.g., tele-pointers, instant messaging, shared editors, etc.) has always been of particular concern to the CSCW (Computer-Supported Cooperative Work) community. Interaction and Collaboration Analysis is a research field that deals with the automatic evaluation of users interactions in order to improve collaborative work processes. These analysis processes usually follow a cycle consisting of three phases: *observation*, *abstraction* and *intervention*. In this context, a current challenge is to design automatic observation mechanisms so that not only can user actions be represented, stored and aggregated by a groupware system, but also so that analysts can identify weaknesses and strengths in the use of the system. In this article, we define an architectural model for the systematic development of a groupware system that incorporates mechanisms to automatically observe users interactions. This approach has been used to develop COLLECE, a groupware system that supports synchronous-distributed collaborative programming.

1 Introduction

Groupware systems are nowadays equipped with multiple mechanisms to support the interaction between the members of a group or the interaction of one user with an object in a shared workspace. Additional mechanisms, commonly based on different visualization techniques, allow the participants in a collaborative task to be aware of the activity where other participants are currently engaged in or have previously carried out. Experience in the development and subsequent use of groupware systems, however, demonstrates that many of them fail to provide useful tools for effective collaboration [15].

The factors for obtaining a successful groupware system are studied by the CSCW (Computer-Supported Cooperative Work) research field, and range from designing and developing appropriate technological solutions to considering the social aspects of an organization [14]. An analysis of how users interact through a groupware system is sometimes carried out [16] in order to obtain different kinds of valuable information such as the misuse, disuse and usage frequency of the different gadgets

provided by a groupware application. This is of special interest in CSCL (Computer-Supported Collaborative Learning), a discipline that studies how information technology can support the collaborative learning of students. In this area, one challenging problem is to design mechanisms to (semi)automatically study the interaction between students when using a CSCL system and the effect of collaboration in their learning. Such an analysis or evaluation task is commonly known as Collaboration and Interaction Analysis in the CSCL community.

There is the additional difficulty when it comes to designing groupware systems, of integrating support to allow the automatic processing of user actions, categorizing them according to certain analysis criteria, storing them and aggregating them in high level variables for later use. In order to resolve this difficulty, this article proposes an architecture that supports the systematic building of groupware systems which integrate support for interaction analysis. According to Phillips [21], this approach can be considered as a reference model because it allows developers to specify the complete structure of a groupware system incorporating interaction observation mechanisms for collaboration analysis at a relatively large granularity. These analysis processes are aimed at improving the communication, collaboration and coordination features of groupware systems.

The article is structured in four additional sections. Section 2 reviews the methods and techniques used in the field of Collaboration and Interaction Analysis in groupware systems. Section 3 presents the framework used to design groupware systems including support for collaboration and interaction analysis. Section 4 defines an architectural model that enables the systematic development of groupware systems that integrate such analysis facilities. Section 5 discusses a set of quality attributes of the architecture proposed. Finally, Section 6 presents the conclusions drawn from the research work and proposes future lines of work.

2 Related Work

Collaboration and Interaction Analysis in groupware systems uses the same information and methods as those used to provide awareness information in groupware systems [4]. This situation is due to the fact that both the analysis processes and the methods to provide awareness information in groupware systems process not only user actions carried out in shared workspaces, but also the actions for coordination or communication so that variables for conceptualizing and representing the users' work may be produced.

In the field of Collaboration and Interaction Analysis in CSCL, Martinez et al. [19] proposed an XML-based language to model the actions carried out by the users of a groupware system. These actions are stored in structured repositories, whereby analysis processes might be performed. However, this proposal does not specify an architecture to support the development of analysis subsystems that process such languages. This point was tackled by the DALIS architecture [20], which contains an analyzer to evaluate user interactions in shared workspaces within a groupware system. Since the DALIS architecture does not incorporate the analyzer in the groupware system, additional modules should be installed to process any interactions carried out and to communicate the interactions to the analyzer.

Although interaction analysis is a research area that arose from a need to analyze students' activities in a CSCL environment, there is a widespread call for analysis processes in several other collaborative systems. This is the case of, for instance, web-based information systems that incorporate mechanisms for evaluating how users use the system [2], systems that define workflows through mining techniques [10], systems that use user histories to improve knowledge management [18], and systems that support and evaluate collaborative software engineering practices [7]. In most cases, these analysis tasks are based on processing log documents which contain users' access to shared workspaces, but the developers do not integrate an analysis subsystem that interacts with other subsystems (e.g., the awareness subsystem).

3 Framework for the Development of Groupware Systems Incorporating Analysis Facilities

One intrinsic difficulty of producing groupware systems is that social protocols and group activities must be considered for a successful design, and these aspects must be supported by software architectures which enable communication and coordination between group participants. There is a wide range of development approaches and, for example, Phillips [21] presents some of the most recognized groupware architectures.

According to Beaudouin-Lafon [1], the complexity of groupware demands great efforts in specifications and development. One main question is how to define models that allow designers to specify and create groupware systems from a semantics-oriented level rather than to start immediately with the implementation. Additionally, a core aim is to define and propose a model-based approach to the design of automatic observation mechanisms that enable user actions to be stored and processed in order to improve quality properties (e.g., usability, interoperability and reusability) of groupware systems.

The research work in this article consists of extending an existing architecture that allows groupware systems to be designed by integrating analysis support of user interaction. The starting point of our approach, is the AMENITIES methodology [11] which enables the design of groupware systems to be addressed systematically and this facilitates subsequent software development. This allows a conceptual model of cooperative systems to be built and focuses on the group concept. It also covers significant aspects of both group behavior (dynamism, evolution, etc.) and structure (organization, laws, etc.). The resulting specification contains relevant information: cooperative tasks, domain elements, person-computer and person-person dialogues, etc. However, it does not include the description of specific observation mechanisms to analyze user actions, i.e. collaboration and interaction between users. These user activity analysis processes consist of the three phases [6] of *observation*, *abstraction* and *intervention*, which are carried out by an evaluator. Each phase has the following aim:

- **Observation:** This phase collects the information necessary to later infer analysis indicators, i.e. analysis significant variables that evaluate specific aspects of the collaborative work processes. In this phase, the evaluator must therefore define what kind of information (typically raw data) needs to be considered for analysis purposes as a basis for calculating indicators and how it may be adequately stored.

- **Abstraction:** This phase uses the information stored in the observation phase to infer analysis indicators.
- **Intervention:** This phase uses the analysis indicators inferred to improve the collaborative work.

In the following section, we present the architectural model proposed in this article and which consists of a small number of named functional elements and the data flow between these elements.

4 Architectural Model for the Development of Groupware Systems Incorporating Analysis Facilities: The COLLECE Case Study

Since groupware applications need to be distributed, it is important to obtain an implementation arising from a set of subsystems that communicate with each other using well-defined interfaces. The architectural design proposal promotes the division/partitioning of the whole system into components (called subsystems) to facilitate its development, reusability, interoperability and maintenance.

According to Garrido et al. [12], a basic groupware application consists of four subsystems: *Identification*, *Metainformation*, *Awareness* and the application itself – in this case, COLLECE, our case study (see Fig. 1).

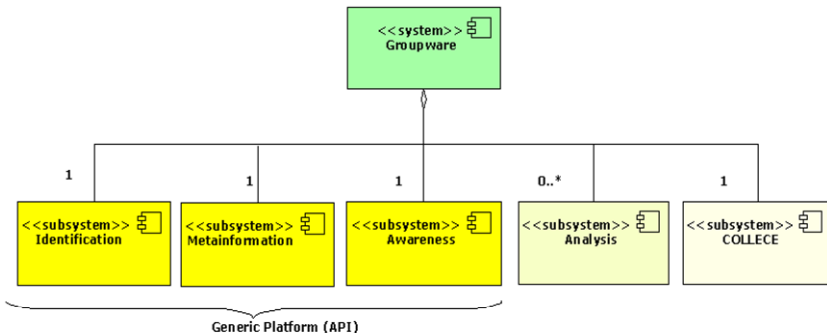


Fig. 1. Architecture for groupware applications: component view

Our architectural design adds an additional *Analysis* subsystem aimed at managing the interaction and collaboration analysis. The *Identification* (user access control), *Metainformation* (metadata management) and *Awareness* (contextual information) subsystems are, in some way or other, always present for every groupware application. The *Analysis* subsystem is only present when analysis is considered and developed in a particular groupware application. Although there is only one *Analysis* or application subsystem in this example, other instances can be easily integrated while this design is maintained.

The COLLECE groupware system [8] supports the collaborative building of computer programs. COLLECE integrates a set of tools to allow users to implement

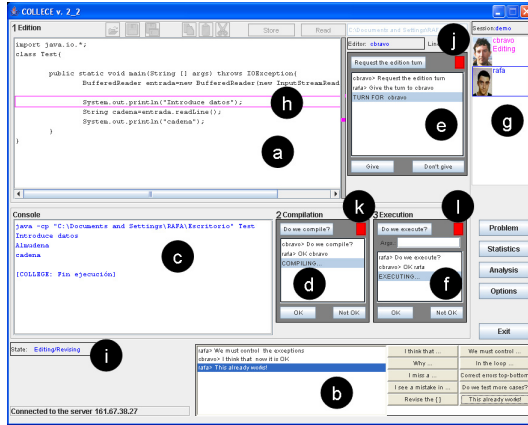


Fig. 2. User interface of COLLECE

programs in a synchronous distributed fashion (see Fig. 2). The system operates on a client-server architecture. The *Identification*, *Analysis* and *Metainformation* subsystems are managed from a central server using Java applications. The *Awareness* subsystem sends information to users and synchronizes workspaces through the JSDT¹ (Java Shared Data Toolkit) toolkit. The system therefore has centralized communication architecture [13].

The COLLECE system is accessed by users through a website. When a user accesses the system, the *Identification* subsystem asks the user for a login and a password. If the entered information is correct, the user can select one of the available working sessions. A working session is defined with information about the requirements of the program to be built and about the programmers who must collaborate. The working sessions are managed by the *Metainformation* subsystem. When the user selects an available session, COLLECE launches the main user interface (Fig. 2), which incorporates the following tools to support synchronous distributed collaborative programming:

- **Code editor** (Fig. 2-a): This is the shared workspace where the program source code is coded.
- **Structured chat** (Fig. 2-b): This tool is used by users to communicate. Although it can be used as a traditional chat, it contains a set of pre-defined communicative acts (e.g., "I think that ...") to make communication more fluid. In order to communicate something to another participant, the user clicks a button containing the label which identifies the communicative act and then simply completes the sentence.
- **Coordination panels:** These tools enable the coordination of program coding (Fig. 2-e), compilation (Fig. 2-d) and execution (Fig. 2-f). The users use buttons in the corresponding panels to request participation in a process and to accept or reject such requests. If a user's request is accepted by all the group members, that user can use either the editor (Fig. 2-a) to code or the console (Fig. 2-c) to compile and execute.

¹ <http://java.sun.com/products/java-media/jsdt/>

The *Awareness* subsystem provides the information to update the shared workspaces, and uses the following mechanisms to enable each user to see each other's work:

- **Session panel** (Fig. 2-g): This contains information about each user: (i) photo, (ii) a description of the user's activity (e.g., editing), and (iii) the user's name in the same color as the user's tele-pointer.
- **Tele-pointers** (Fig. 2-h): These are marks (colored rectangles) in the editor that highlight the line code being edited by a user.
- **State label** (Fig. 2-i): This text describes the group's work phase (edition, compilation, execution).
- **Semaphores** (Fig. 2-j,k,l): These "lights" are incorporated into the coordination panels. The possible colors are green for when there is a request that should be answered by the user and red for when there are no requests pending in the panel.

COLLECE integrates an analysis subsystem that automatically processes user activity. In order to achieve this, a user playing the role of evaluator defines what activities should be processed. The analysis subsystem then accesses the information provided by the *Awareness* and *Metainformation* subsystems to observe user activity.

In the following subsection, a functional view will show the functional relations between each subsystem with the others through interfaces, while a behavioral view describes the collaboration between subsystems in order to automatically process user activity.

4.1 Functional View

Figure 3 shows the five subsystems and their use relations on the basis of the associated functionality. Each subsystem provides an interface designed to achieve independence

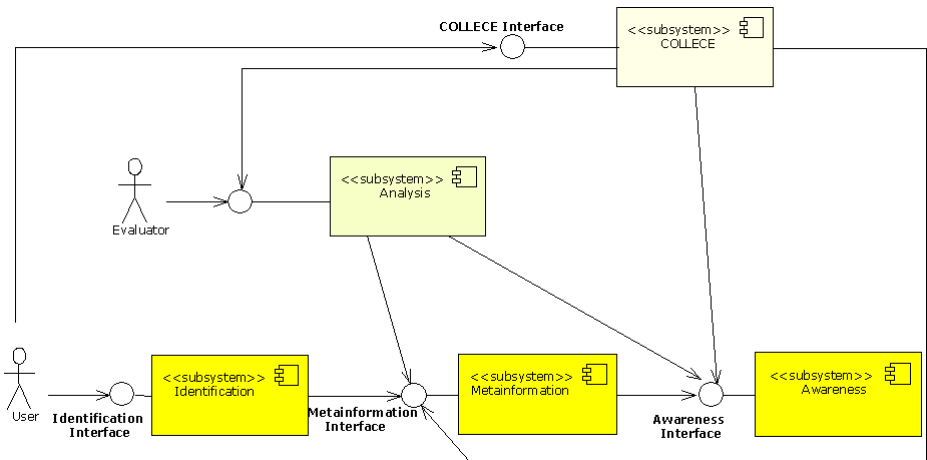


Fig. 3. UML interface diagram for the functional view

between the subsystems and other applications using them. In particular, the *Identification* subsystem is used to start the applications and to control users' access to the system. The *Metainformation* subsystem supports all the functionality for managing metadata (new roles acquired, tasks for each role, etc.) specified in the cooperative model derived from the application of the AMENITIES methodology. The *Awareness* subsystem is intended to provide contextual information for analysis mechanisms and user interface components (list of participants, tele-pointers, etc.) which are in charge of implementing the group awareness required by participants for effective collaboration. The *Analysis* subsystem provides all the necessary services for collaboration and interaction analysis which are of interest to the evaluator.

4.2 Behavioral View

System behavior is specified on the basis of the functional structure (described in the previous subsection) using an interaction diagram. In Figure 4, a UML sequence diagram describes the interactions between the subsystems and the actors (two users and an evaluator) and between the different subsystems themselves by focusing on the sequence of messages exchanged.

Firstly, users must register in the system so that other users can choose to observe them (*combined fragment par* in Figure 4). The evaluator then starts an interaction analysis process; as of this moment, any action carried out by any of the users is recorded by the analysis subsystem, as shown in Figure 4. The diagram also expresses an interaction sample which consists of a compilation request which is recorded by the *Analysis* subsystem.

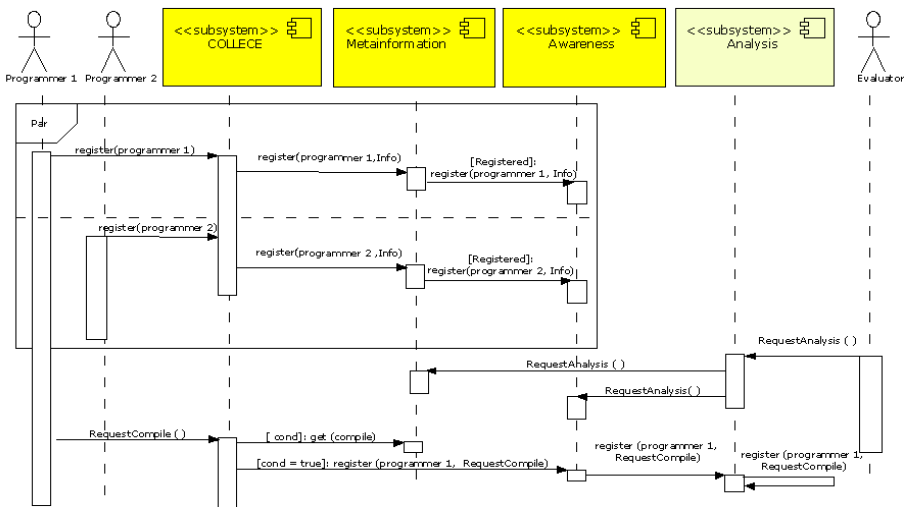


Fig. 4. UML sequence diagram for the behavioral view

5 Discussion: Software Architecture and Work in Progress

Having defined a software architecture that allows groupware systems to be designed which incorporate an analysis subsystem, it is necessary to evaluate the quality attributes of such an architecture. For this purpose, evaluation was conducted by studying a series of quality attributes defined in the IEEE 610.12-1990 standard [17] based on various experiments [3]. During the evaluation process, COLLECE usability was assessed and this usability [5] could be improved by adapting the following tools:

- **Structured chat:** Using the COLLECE observation mechanisms, it was possible to identify which pre-defined messages in the chat were not used by the users. In doing so, the user interface was adapted to include a new set of communication messages.
- **Coordination panels:** The observation mechanisms also enabled those coordination panels where there was often disagreement to be identified. In such cases, the efficiency of the collaborative work was improved by adapting the coordination policies, which were changed so that a user requesting access to a shared workspace only needs to be accepted by half the partners.

A second set of experiments are also being conducted to validate the quality attributes of the software architecture. These attributes have properties that enable developers to make decisions regarding the design of the groupware systems. These properties and evaluation experiments are as follows:

- **Reusability:** This is *the degree to which a software module or other work product can be used in more than one computer program or software system*. The architecture comprises a set of independent subsystems that interact with each other through the functionalities defined in the API, and this enables subsystems to be reused without having to change new groupware systems. This situation is being validated by integrating the COLLECE analysis subsystem into other groupware systems, e.g., SPACE-DESIGN [9].
- **Interoperability.** This is *the ability of two or more systems or components to exchange information and to use the information that has been exchanged*. The analysis subsystem requests information from the *Awareness* subsystem in the architecture proposed. This communication between subsystems allows the information needed to achieve the goal of the analysis task to be chosen. In particular, COLLECE supports analysis tasks with the following goals: (i) to analyze interactions with the user interface [5], (ii) to implement an automatic evaluation of the collaborative process carried out by the users and to show analysis indicators [6], and (iii) to compare productivity between practices based on Agile Software Methodologies and traditional practices [8].

6 Conclusions

Groupware systems support the collaborative work of users who share artifacts and documents in workspaces in a coordinated way. In this context, analysis processes can be defined so that collaborative work may be characterized (thereby enabling

different kinds of intervention to be established) and therefore improved. The analysis subsystems usually make use of the information managed by other subsystems of the groupware system. However, so far there exists a lack of proposals that guide developers to integrate analysis subsystems in groupware architectures in order to interact with other subsystems.

This article proposes an analysis subsystem which incorporates mechanisms to automatically observe collaboration and interaction between users. This analysis subsystem has been integrated into a groupware architecture intended to guide the development of groupware systems.

This architecture is used as a reference model that allows developers to define the whole structure of a groupware system. The architecture is made up of subsystems that interact to automatically process user activity. The architecture has been validated in the development of the COLLECE groupware system. COLLECE makes use of the aforementioned analysis subsystem that automates the observation of user activity. In particular, the analysis subsystem was used in several experiments that have allowed the quality attributes of the architecture to be evaluated. In addition, the COLLECE analysis subsystem has enabled COLLECE usability to be improved.

In the future, this architecture will be used to develop mechanisms that automate not only the observation phase, but also the abstraction and intervention phases of an analysis life cycle.

Acknowledgments

This research is supported by the Comunidad Autónoma de Castilla - La Mancha (Spain), Consejería de Educación y Ciencia, under the coordinated project PAC07-0020-5702.

References

1. Beaudouin-Lafon, M. (ed.): In: Computer Supported Co-operative Work. Ed. Trends in Software, vol. 7. John Wiley & Sons, Chichester (1999)
2. Billeskov, K., Simonsen, J.: Http Log Analysis: An Approach to Studying the Use of Web-Based Information Systems. *Scandinavian Journal of Information System* 16(1), 145–174 (2004)
3. Bosch, J.: Design & Use of Software Architectures – Adopting and evolving a product-line approach. Pearson Education, London (2000)
4. Bratitsis, T., Dimitracopoulou, A.: Monitoring and Analysing Group Interactions in asynchronous discussions with the DIAS system. In: Dimitriadis, Y., Zigurs, I., Gomez-Sanchez, E. (eds.) CRIWG 2006. LNCS, vol. 4154, pp. 54–61. Springer, Heidelberg (2006)
5. Bravo, C., Duque, R., Gallardo, J., García, J., García, P.: A Groupware System for Distributed Collaborative Programming: Usability Issues and Lessons Learned International. In: Van Hillegersberg, J., Harmsen, F., Amrit, C., Geisberger, E., Keil, P., Kuhrmann, M. (eds.) Workshop on Tools Support and Requirements Management for Globally Distributed Software Development. CTIT Workshop Proceedings, Centre for Telematics and Information Technology, pp. 50–56 (2007)

6. Bravo, C., Redondo, M.A., Verdejo, M.F., Ortega, M.: A Framework for the Analysis of Process and Solution in Synchronous Collaborative Learning Environments. *International Journal of Human-Computer Studies* (in press)
7. Cook, C., Churcher, N.: Modelling and Measuring Collaborative Software Engineering. In: Estivill-Castro, V. (ed.) *Proc. Twenty-Eighth Australasian Computer Science Conference (ACSC 2005)*, Newcastle, Australia, CRPIT, vol. 38, pp. 267–276. ACS (2005)
8. Duque, R., Bravo, C.: Analyzing work productivity and program quality in collaborative programming practices. In: *The 3rd International Conference on Software Engineering Advances, ICSEA 2008*, IEEE Computer Society Press, Los Alamitos (2008) (in press)
9. Duque, R., Gallardo, J., Bravo, C., Mendes, A.J.: Defining tasks, domains and conversational acts in CSCW systems: the SPACE-DESIGN case. *Journal of Universal Computer Science* 14(9), 1463–1479 (2008)
10. Gaaloul, W., Alaoui, S., Baina, K., Godart, C.: Mining Workflow Patterns through Event-Data Analysis. In: *Proceedings of the 2005 Symposium on Applications and the Internet Workshops*, pp. 226–229 (2005)
11. Garrido, J.L., Gea, M., Rodríguez, M.L.: Requirements Engineering in Cooperative Systems. In: *Requirements Engineering for Sociotechnical Systems*, pp. 226–244. IDEA GROUP, Inc., USA (2005)
12. Garrido, J.L., Padereswki, P., Rodríguez, M.L., Hormos, M.J., Noguera, M.: A Software Architecture Entended to Design High Quality Groupware Applications. In: *Proc. of the 4th International Workshop on System/Software Architectures (IWSSA 2005)*, Las Vegas, (USA) (2005)
13. Greenberg, S., Roseman, M.: Groupware toolkits for synchronous work. In: Beaudouin-Lafon, M. (ed.), John Wiley & Sons, Chichester (1996)
14. Grudin, J.: Why CSCW applications fail: problems in the design and evaluation of organization of organizational interfaces. In: *Proceedings of the 1988 ACM Conference on Computer-Supported Cooperative Work. CSCW 1988*, pp. 85–93. ACM Press, New York (1988)
15. Grudin, J.: Computer-supported cooperative work: History and focus. *IEEE Computer* 27(5), 19–26 (1994)
16. Gutwin, C., Penner, R., Schneider, K.: Group Awareness in Distributed Software Development. In: *Proc. ACM CSCW 2004*, pp. 72–81 (2004)
17. IEEE std 610.12-1990 (n.d.): IEEE Standard Glossary of Software Engineering Terminology (Retrieved January 19, 1990)
18. Komlodi, A., Lutters, W.G.: Collaborative use of individual search histories. *Interact. Comput.* 20(1), 184–198 (2008)
19. Martínez, A., Dimitriadis, Y., de la Fuente, P.: Towards an XML-based model for the representation of collaborative action. In: *Proceedings of the Conference on Computer Support for Collaborative Learning (CSCL 2003)*, Bergen, Norway, pp. 14–18 (2003)
20. Muehlenbrock, M., Hoppe, U.: Computer supported interaction analysis of group problem solving. In: *Computer Support for Collaborative Learning*, Palo Alto, USA, pp. 398–405 (1999)
21. Phillips, W.G.: *Architectures for Synchronous Groupware Technical Report 1999-425* (1999); ISSN 0836-0227-1999-425

A Case Study on Architectural Maturity Evaluation: Experience in the Consumer Electronics Domain

Kangtae Kim

Samsung electronics, Software center, maetan 3dong, Suwon, Korea
Kangtae.kim@samsung.com

Abstract. This paper introduces our experience of applying an architecture evaluation model to enhance quality attributes of product line in the consumer electronics field. We proposed an evaluation model for architecture design and performed assessment on several platforms to evaluate reusability and maintainability as a core asset of product line. Former researches such as PULSETM, FEF(Families Evaluation Framework), and PLPF(Product Line Practice Framework) provide the whole spectrum of building, deploying and maintaining software product lines based on the BAPO(Business, Architecture, Process and Organization). But we focused on software architecture design itself further because we are mainly concerned with design and implementation issues thus concentrating on the architecture criteria. In this paper, we will provide the practitioners with an experience of our approach, core concept and lessons learned relies on several case studies on design practices.

Keywords: SW product line, SW architecture, architecture maturity.

1 Introduction

A practical, systematic approach for improving and maintaining product line development practices is essential to the success of populating the product line paradigm to the organization. It supports practitioners in analyzing their current development practices, in developing a product line target situation, and in identifying a migration path [1]. Former researches such as PuLSETM(ProdUct Line Software Engineering)[2], FEF(Families Evaluation Framework)[3][4], PLPF(Product Line Practice Framework)[5] are well defined frameworks for evaluating and improving product line development practices. They are based on BAPO(Business, Architecture, Process, Organization) model to cover whole spectrum of software product line practices. But we are mainly concerned with architectural quality attributes especially on reusability and maintainability thus specializing evaluation models focused on architecture design. The approach is based on the assumption that most of software architectures which derive software platform in consumer electronics share common but abstract goal for reuse. It relies on five attributes: design of commonality/variability, layered

architecture, standard interface, abstraction mechanism and documentation. These five attributes have close inter-relationships between them thus assessed in a correlative manner.

We assessed about 30 platforms that are currently productized based on this model. Each platform is assessed twice over the duration of 6 months to verify availability of evaluation model as a tool for improving software product line development practices. By performing the first assessment, we have found several drawbacks in architecture designs and corresponding implementations. After improvement activities on each of concern, we performed the second assessment for verification.

This paper is outlined as follows. The next section shows the structure of the architecture evaluation model we defined. A case study, lessons learned and conclusion with future work follow it.

2 Structure of the Architecture Evaluation Model

In this section we will show the overall structure of the architectural evaluation model and detailed evaluation criteria of each view.

2.1 Architecture Evaluation Model

FEF and PuLSETM are structured in five levels in common. FEF suggested an architectural dimension for evaluating architecture of product line and PuLSETM classified practices into more detailed levels such as scoping, modeling, architecting, designing, and so on. On the other hand, our proposed model is structured in a non-scale manner for simplicity of application. We adopted several practices of level 2 to 5 of FEF and architecting, designing, coding and instantiating of PuLSETM framework in a flat structure. Several architecture design guide lines are also adopted for completeness [6][7]. Selected practices are classified into 5 platform attributes which have been developed for corporate design guidelines, i.e., commonality/ variability, layered architecture, standard interface, abstraction mechanism and documentation [8].

2.2 Commonality / Variability

A software platform should be configurable to adapt variations of target products with selective components. It should be analyzed in terms of the commonality and variability of features and designed accordingly to be applied to the target products. The platform should be configurable to adapt variation of products with selective components as well. In many cases, platform is developed based on a limited short-term product planning. It brings frequent changes in the architecture and assets themselves, which results in poor reuse of platform finally. The detailed items of the criteria for the commonality and variability attribute are shown in Table 1.

Table 1. Criteria of commonality and variability

Classification	Items
Definition of commonality & variability	Does it define the scope of the platform coverage?
	Does it define the criteria for commonality and variability with respect to functionality, environment and technical issues?
	Are some techniques for variability design applied?
	Does it include the feature of the expecting future product?
Design of commonality & variability Implementation of commonality & variability	Is it possible to visualize commonality and variability at the top-level platform architecture?
	Is there an applied method to present commonality and variability in design phase?
	Are some design techniques (style/pattern) to manage and extend commonality and variability applied?
	Are some techniques such as limitation of access to keep commonality in code level applied?
Traceability of commonality & variability	Are some techniques such as modification of the code directly, plug-gable mechanism with component, configuration tool to handle variability in code level applied?
	Is it possible to trace variability from requirement phase to design phase?
Definition of commonality & variability	Is it possible to trace variability from design phase to implementation phase?
	Is there a definition of the scope of the platform coverage?

2.3 Layered Architecture

A software platform should be designed in a layered fashion to enhance maintainability and reusability with a separation of concerns and concentration of functionalities. This reduces complexity and enhances the maintainability of a platform. The detailed items of the criteria in the layered architecture attribute are shown in Table 2.

Table 2. Criteria of layered architecture

Classification	Items
Definition of layered architecture	Is there a criterion for separating an architecture into layers?
	Is the role of each layer defined?
Design of layered architecture	Are the relationships between layers well defined?
	Are there any access rules applied between layers?
Implementation of layered architecture	Are there some techniques or rules to manage layered architecture? (for example, back-calls and skip-calls)

2.4 Abstraction Mechanism

A software platform should be designed with consideration of a variety of target environments such as the OSs, device drivers and chipsets. Embedded software in the consumer electronics domain is highly influenced by its execution environments and these change frequently from projects to projects. It is a critical success factor to build a platform that can be transparent to target environment. The detailed items of the criteria for the abstraction mechanism are shown in Table 3.

Table 3. Criteria of abstraction mechanism

Classification	Items
Definition of abstraction mechanism	Are the factors that influence the platform analyzed?
	Are there factors that are expected to influence the platform in the future, such as changes in hardware, operating systems, device drivers or graphic user interfaces?
Design of abstraction mechanism	Is there an abstraction mechanism that is employed to describe the platform's architecture?
	Are there some rules to maintain the abstraction mechanism?
Implementation of abstraction mechanism	Is the abstraction mechanism applied at the implementation level?
	Are the rules of the abstraction mechanism monitored and maintained?

2.5 Standard Interface

A software platform should define standard interfaces. A standard interface regulates the use of platform components and provides opportunities to reuse them in a pre-defined way. It supports configurability of a software platform. It is one of effective ways to maintain the layers of a platform architecture and abstraction mechanisms. The detailed items of the criteria for the standard interface are shown in Table 4.

Table 4. Criteria of standard interface

Classification	Items
Definition of standard interface	Are there standard interfaces defined between layers and components?
	Are there rules for interface standardization? For instance, - Interface documenting guide - Naming convention
Design of standard interface	Are inner functions and outer functions (interface) separated?
	Are interface parts separated from their implementation parts?
Implementation of standard interface	Are function calls between components made with only standard interfaces (i.e., no direct access)?

2.6 Documentation

A software platform is mainly used by software developers. Consequently, it is very important to help these developers to understand the design concept, architecture design, porting and performance issues, API, and others. We prescribe that, as a core requirement, appropriate documentations should be produced that reflect on the four guidelines we have dealt with above. In a production project, most documents can be derived from platform documents and reused.

2.7 Rating Scale

Our rating scale is defined to measure the level of each attribute. Each attribute is assessed on a N(Not achieved)-P(Partially achieved)-L(Largely achieved)-F(Fully achieved) rating scale with the score result(0~100). Table 5 shows details.

Table 5. Rating scale

Level	Criteria
Not Achieved	Few practice. Achievement of practices is under 25%
Partially Achieved	Some practice, but many weak points. Achievement of practices is between 25% ~ 50%
Largely Achieved	Many practice, but some weak points. Achievement of practices is between 50% ~ 85%
Fully Achieved	Best practice and meet the goal. Achievement of practices is over 85%

Table 6. Metrics for each attribute

Attribute	Metrics
Commonality / variability	<ul style="list-style-type: none"> - # of communized(standardized) assets - # of reused assets * applied on requirements, components, test cases ** derived metrics are to be combination of metrics
Layered architecture	<ul style="list-style-type: none"> - # of dependency b/w layers(function call, data, header file dependency) - # of violation to architecture design(skip calls, back calls)
Abstraction mechanism	<ul style="list-style-type: none"> - # of abstractions of target : OS, hardware drivers, UI - # of abstracted feature set(interfaces) * coverage of abstraction for target and feature set is essential for completeness
Standard interface	<ul style="list-style-type: none"> - # of standardized interfaces - # of dependency b/w each components, sub systems against architecture - # of violation to architecture design(direct calls) * Coverage of standard interface (should be vs. as is) is essential and violation cases are critical
Documentation	<ul style="list-style-type: none"> - # of documents against architecture decomposition rule

2.8 View and Metric

Most important aspect of the evaluation model is views and metrics which harmoniously indicate the status of architectural maturity. In the section 2, each maturity criteria is described in qualitative way. Quantitative indicators are also required for formal assessment. View and metrics are categorized in platform attribute which is introduced in the section 2 with relationship with standard software metric ISO 9126.

3 Case Study

Over 30 platforms in around 10 divisions are assessed against this model this year as shown in Table 6. Each platform is assessed twice for verification of usefulness. In each platform’s point of view, the evaluation result of the assessment is a collection of drawbacks and improvement items at the same time. A summary of the application will be shown in the rest of this section. Fig. 1 shows summary of improvements in terms of the platform attribute.

Major improvement items on commonality and variability attribute are to

- 1) Define a set of criteria for identifying commonality and variability using some requirement analysis techniques like feature modeling [9].
- 2) Design and visualize commonality and variability at the top-level platform architecture.
- 3) Manage the traceability of variable elements from requirement phase to design phase and create a management system for that.

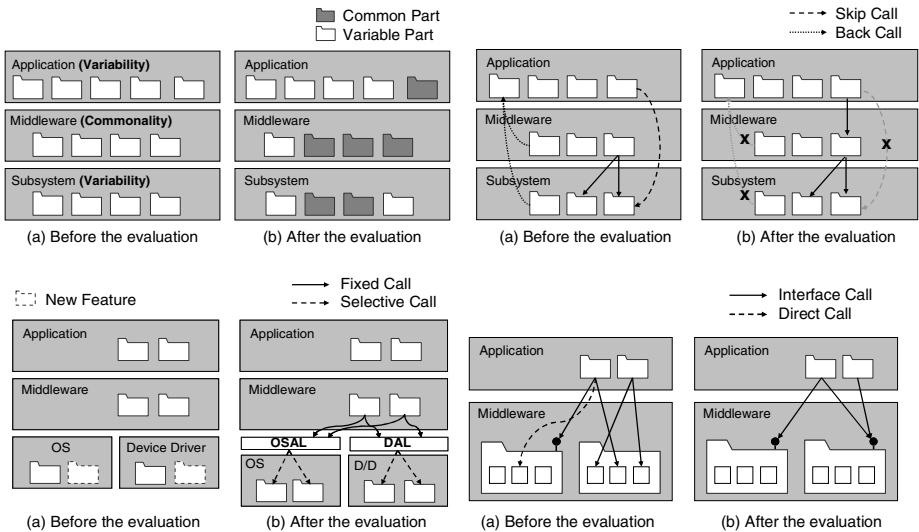


Fig. 1. Summary of improvement on each platform attributes

In the area of layered architecture attribute, we have removed all of the violation cases based on the layered architecture access policy. Major improvement items are to

- 1) Re-design and re-structure the layered architecture in question according to the role of each layer and the relationship between them.
- 2) Remove accesses that violate the layered architecture access policy, such as back calls and skip calls [12].
- 3) Ensure the consistency between the logical architecture design and implemented directory structure.

With respect to abstraction mechanism, we have newly created the OS abstraction layer (OSAL) and Device driver abstraction layer (DAL) under the middleware layer. Major improvement items on the abstract mechanism are to

- 1) Create an OS abstraction layer that is helpful in avoiding the direct influence from change of OS types.
- 2) Create a device driver abstraction layer for accommodating various device drivers.
- 3) Create a hardware abstraction layer for accommodating various chipsets.

In the area of standard interface, we have elicited a few standard interface violations. Major improvement items on the standard interface are to

- 1) Make standard interface rules like a naming convention and interface documenting guide.
- 2) Re-structure standard interfaces from the existing function calls between layers.
- 3) Impose a standard interface access policy by removing direct accesses that does not conform to the standard interface.

Major improvements of the documentation are the reflection of the four attributes above in our architecture evaluation model onto corresponding platform documentations. Major improvement items on the documentation

- 1) Present commonality and variability attribute on the related platform documentations such as requirement, architecture design and test documents.
- 2) Ensure consistency between design and implementation, and corresponding documentations.
- 3) Define and use some specific notations and views for describing platform architectures.

In this case study, we discovered that our architectural evaluation model is very helpful for platform software developers in the consumer electronics domain to improve their platform architectures by using the criteria of our architectural evaluation model as a technical improvement guideline.

As a result of improvement activities on each platform after first assessment, It is verified that average 20% of maturity has been improved through whole spectrum of attributes in the second assessment. Documentation attribute is most significantly improved area. It is because there's no or little document maintenance effort in the product group in general. Layered architecture attribute is also improved through elimination of call violation such as skip calls or back calls. Platforms are cleaned up in terms of interface during improvement activities and it results in clear layers and well defined interfaces . Variability for derivation is also identified using feature

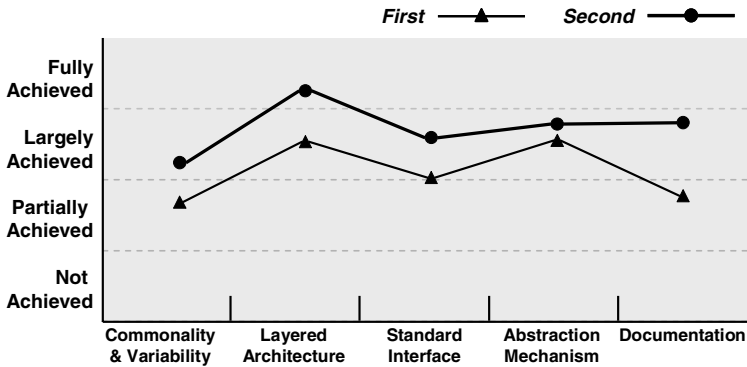


Fig. 2. Summary of the assessment results

analysis techniques and applied to design and implementation. Code configuration for variability is also empowered using several build configuration automation techniques.

As shown in figure 2, the actual results of the second assessment using our architecture evaluation model are notably improved compared with the first assessment results.

4 Lessons Learned

The following is what we have learned from establishing and adopting an architectural evaluation model. Regarding designing the evaluation model and performing assessment,

1. Component evaluation also needed for practical and effective assessment.

The evaluation model provides several views to assess the maturity of the architecture design of a software platform. It focuses on high-level design of architecture with respect to platform attributes. But there is lack of consideration on maturity of building blocks of platform on implementation level. Components are basic building blocks of a software platform, thus maturity of components is closely related with the quality attributes of a platform. It seems to be essential to extend an architecture evaluation model to cover implementation by adopting a component maturity model.

2. Improvement items should come from assessment results that reflect the status of each development domain.

Organizations and even their platforms have their own development environment and domain constraints in nature. Performance can be one of major constraints in considering alternatives against evaluation criteria. Such constraints that are endowed by the stakeholders might have the highest priority. If there are conflicts between constraints and evaluation criteria and constraints have a higher priority, it is supposed that one should adopt the higher priority constraints i.e. those that can go against the evaluation criteria intended. In this case, improvement items that are derived from the assessment would be ignorable and should be excluded from the assessment result.

3. Improvement items should be moderated with development parties.

Quantitative analysis performed by automated code analysis tools indicates formal and objective status of architecture design. But a quantitative approach is not quite enough. An additional qualitative, analogical approach is needed because numbers cannot be transformed into improvement items directly. Our evaluation model insists a normalized evaluation result which is hard to reflect specific environment and constraints of each platform. What is more, an evaluation model itself is possibly not complete enough to find all aspects of drawbacks. An arbitrating activity on the assessment result with development parties is quite important for deriving practical improvement items. We have good references on it such as CMMI[10], IS15504[11], so that it might be helpful to adopt them with regard to collecting improvement items.

5 Conclusion and Future Work

Software product lines have been recognized as a software development paradigm that leads to improvements in terms of software cost, productivity, and quality. In this paper, we presented an architecture centric design evaluation approach for product line development and its application to families of consumer electronics software at Samsung Electronics. The paper analyzed in retrospective the results and experiences in evaluating product line development practices. The architecture evaluation model introduced in the paper aims both to support the transition to a product line approach by transforming legacy architecture to a product line based one as well as the continuing improvement of the legacy architecture. The evaluation criteria works as design guidelines of software product line for enhancing reusability, extensibility and maintainability as a principle of design and implementation. The case study shows the effectiveness of the model such that most platforms are improved in their capability by the items found in the assessment.

As a future work, we plan to refine and extend the evaluation criteria to cover component level design practices. The architectural view will be integrated in coordination with a component level view for evaluating detailed designs and implementation practices.

References

1. Schmid, K., van der Linden, F.: Improving product line development with the families evaluation framework(FEF). In: Proceedings of the 11th International Software Product Line Conference, Kyoto, Japan, September 10-14 (2007)
2. Schmid, K., Widen, T.: Customizing the PuLSETM Product Line Approach to the Demands of an Organization. In: Proceedings of the 7th European Workshop on Software Process Technology, February 21-25, 2000, pp. 221–238 (2000)
3. van der Linden, F., Schmid, K., Rommes, E.: Software Product Lines in Action, pp. 79–108. Springer, Heidelberg (2007)
4. Schmid, K.: A comprehensive product line scoping approach and its validation. In: Proceedings of the 24th International Conference on Software Engineering, Orlando, Florida, May 19-25 (2002)

5. Clements, P., Northrop, L.: A Framework for Software Product Line Practice, Version 3.0 Pittsburgh, PA: SEI, Carnegie Mellon University (September 2000), <http://www.sei.cmu.edu/plp/framework.html>
6. Bosch, J.: Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach. Addison-Wesley, New York (2000)
7. Clements, P., Northrop, L.: Software Product Lines: Practices and Patterns. Addison-Wesley, New York (2002)
8. Kim, K., Kim, H., Kim, W.: Building software product line form the legacy systems, experience in the digital audio & video domain. In: Proceedings of the 11th International Software Product Line Conference, Kyoto, Japan, September 10-14, 2007, pp. 171–180 (2007)
9. Kang, K.C., Kim, M.Z., Lee, J.J., Kim, B.K.: Feature-Oriented Re-engineering of Legacy Systems into Product Line Assets – a case study. In: Obbink, H., Pohl, K. (eds.) SPLC 2005. LNCS, vol. 3714, pp. 45–56. Springer, Heidelberg (2005)
10. Chrissis, M.B., Konrad, M., Shrum, S.: CMMI(R): Guidelines for Process Integration and Product Improvement, 2nd edn. The SEI Series in Software Engineering. Addison-Wesley, Reading (2003)
11. Emam, K.E., Drouin, J.-N., Melo, W.: SPICE: The Theory and Practice of Software Process Improvement and Capability Determination. Wiley, Chichester (2002)
12. Sarkar, S., Rama, G.M., Shubha, R.: A Method for Detecting and Measuring Architectural Layering Violations in Source Code. In: 13th Asia Pacific Software Engineering Conference (APSEC 2006), pp. 165–172 (2006)

Evaluation of an Agent Framework for the Development of Ambient Computing

Marcela D. Rodríguez¹ and Jesús Favela²

¹ Facultad de Ingeniería, Autonomous University of Baja California, Mexicali, México

² Ciencias de la Computación, CICESE, Ensenada, México
marcerod@uabc.mx, favela@cicese.mx

Abstract. We have proposed to use software agents as a basis for implementing ambient computing systems. We developed the SALSA middleware, which enables the creation of autonomous agents reactive to the context of ambient computing environment; allows to communicate agents with other agents, users and services; and to implement agents that represent users, act as proxies to environment resources, or wrap a complex system's functionality. We have evaluated SALSA based on empirical studies for assessing infrastructures used for implementing interactive systems. This evaluation included in-lab programming experiments and design exercises to assess the facilities provided by SALSA agents. We present evidence that SALSA is sufficiently powerful to facilitate the implementation of ambient computing services.

Keywords: ubiquitous computing, autonomous agents, middleware.

1 Introduction

Ubiquitous computing or ambient computing suggests new paradigms of interaction and collaboration between users and/or services, inspired by widespread context-aware access to information and computational capabilities, which have led developers of ambient computing systems to face several challenges in order to cope with the complexities associated with the development of these systems. Such challenges have major implications for the software infrastructure that must facilitate the progressive development of an ambient computing system. Among these complexities are those due to the heterogeneity in computing, communication and devices embedded in the physical environment that lead to challenges related to the adaptation of information by taking into account the user's context. Another source of complexity is scalability with respect to devices, people, and time since new devices and people can join the environment at any time. Finally, providing software support for building ambient computing systems (such as toolkits, frameworks and middleware) is a major challenge identified by the ubiquitous computing research community [1][2]. However, the existing development architectures provide support for dealing with some of the complexities of ubiquitous computing systems, but they do not address how to facilitate the evolution of a ubiquitous computing system. The ActiveSpaces project developed the Gaia meta-operating system, which is a distributed middleware infrastructure that coordinates software entities and heterogeneous networked devices contained in a

physical space [3]; and One.world enables the development of adaptable pervasive applications and the discovery of resources [4]. Developers have also used software agents as a technique for developing complex distributed software systems. For instance, some projects use the agents' paradigm to speed-up concurrent processing, and provide more reliability because of the lack of a single point of failure and improve the responsiveness of the system. However, they do not use autonomous agents as an abstraction tool for the design and construction of these systems [5][6][7]. A software agent is a software entity that acts on behalf of someone to carry out a particular task which has been delegated to it. To do this, an agent might be able to take into account the peculiarities of users and situation. Each agent might possess a greater or lesser degree of attributes, such as: autonomy (to act on their own), reactivity (to respond to changes in the environment), pro-activity (to reach goals), cooperation (with other agents to efficiently and effectively solve tasks) and adaptation (to learn from its experience). The motivation of our research work to explore the use of autonomous agents for creating ambient computing systems is that they possess the characteristics of distribution, reactivity, collaboration and adaptation of their artifacts, thus sharing several characteristics with agents. To enable developers to create the software entities of an ambient computing environment with which users need to seamlessly interact, the Simple Agent Library for Smart Ambients (SALSA) was created [9]. The methodology for exploring the use of autonomous agents for creating ambient computing systems and for creating SALSA, consisted of several iterative phases. The first phase consisted of selecting scenarios that illustrate how ubiquitous computing technology enhances users' activities. These scenarios then were analyzed to identify how autonomous agents can be used for designing ambient computing systems. From these analyses we identified several design issues to consider for abstracting the requirements of the SALSA middleware [9]. In the next phase the SALSA middleware was designed and implemented. Then, SALSA was evaluated. Before presenting the results of this evaluation, we first present in Section 2 the SALSA agent design issues and requirements. Section 3 explains the SALSA middleware. Section 4 and 5 presents the results of evaluating SALSA. And finally, Section 6 presents our conclusions and future work.

2 Autonomous Agents for Designing Ambient Computing Systems

The selected usage scenarios of ambient computing systems, illustrate how users can interact with other users, services and devices available throughout the environment, through messages that are delivered when certain contextual conditions are met. From these scenarios, we identified the following design issues regarding the functionality of autonomous agents for creating ubicomp environments [9]:

- Autonomous agents are decisions makers that would review the context and make decisions about what activities to do, when to do them, and what type of information to communicate to whom.
- Autonomous agents are reactive to the contextual elements of the environment. For this, agents need mechanisms to perceive, recognize and disseminate different types of context information.

- Autonomous agents can represent users, act as proxies to information resources, services and devices, or to wrap a complex system's functionality.
- Autonomous agents should be able to communicate with other agents, or directly to users and services. For this, agents need a platform of communication that enables them to convey information to other agents, users, and information resources by using the same protocol of communication.
- Agents need a communication language that enables them to convey different type of messages, such as messages for negotiating services offered by other agents, requesting information from devices or services, and responding to such requests. The communication platform should enable users to be aware of the presence of other users and agents that offer relevant services for them.
- Autonomous agents may have a reasoning algorithm as complex as the logic of its functionality. For this, agents may need a reasoning algorithm which may include a simple set of rules or a more complex reasoning.

3 The SALSA Middleware

The SALSA middleware mainly consists of a Communication Platform and an API (set of abstracted classes). The communication platform consists of a communication channel among agents and users, which is a Broker component that is responsible for coordinating the communication. The Broker also enables the configuration of the environment in a manner that is transparent for users since they do not know the location of the agents even though they can be aware of the presence of some of these agents. The implementation of the Agent Broker is the Jabber Instant Messaging and Presence (IM&P) server. SALSA also provides a protocol of communication which consists of an expressive language (based on XML) that enables the exchange of different types of objects between agents (such as perceived information, requests for services), between agents and users (such as events generated by the user's actions), and between agents and services (such as the state of a service). This information will be sent or perceived by the agent through a proxy to the Broker, which is an agent's component created by developers by using the SALSA API. The SALSA API is a class framework designed for facilitating the implementation of the agents' components for perceiving, reasoning and acting, and to control the agent's life cycle. The SALSA middleware provides an Agent Directory service which is accessible through the Initialize and Register module of the SALSA API.

3.1 SALSA API

As depicted in figure 1, the set of classes provided by SALSA enable the implementation of the main components of an agent for perceiving, reasoning and acting. Two types of perception can be implemented with SALSA: active and passive. The passive perception was implemented based on the Observer design pattern. This type of agent perception starts when a user, device or other agent sends data to an agent through the Agent Broker. In this case an agent has the role of observing the environment and acting according to the information received. The `PassiveEntityToPerceive` class represents the subject to be observed by the agent; and the `PassivePerception` class captures the information sent by

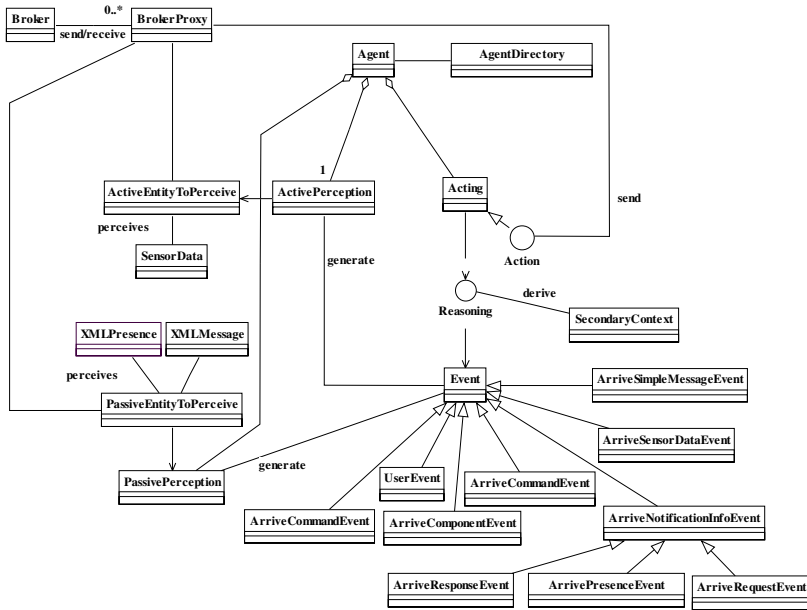


Fig. 1. SALSAs Architecture

the subject. The passive perception of a SALSAs agent, in which data is received through its Broker Proxy (an IM client), is due to another agent that sends information by using the communication methods of the SALSAs API. For the active perception, an agent decides on its own when to sense an environment entity. This type of perception implements the Adapter design pattern. The only active perception supported by SALSAs, is when the agent perceives data directly from a sensor or device. When any of the perception components receive information, a SALSAs event is generated indicating the type of information to the reasoning component. A SALSAs event contains the perceived data. The information perceived by an agent is subtracted from the event by the reasoning component in order to be analyzed. The Reasoning class contains the abstract method `think()` that should be implemented by the developer by using a reasoning algorithm, such as a simple condition-action rule or a neural network. The reasoning component can use the facilities of SALSAs to derive context information from the primary context information perceived by an agent. For this, SALSAs provides the class `DeriveContext` which uses an XSL file as a filter in which the developer specify a set of rules to deduce secondary context from the data perceived by the agent. The derive context component returns an XML message to the agent's reasoning. To implement the action component, the framework provides the `Action` class with an abstract method that a developer should overwrite to specify how the agent must react. From the action component, the agent can invoke the methods of communication provided by SALSAs in order to collaborate with other agents. These methods facilitate the composing, sending, and receiving of messages between agents. However, the code for every content message type of the communicative

act is left to the programmer, because it depends on the intent of the message generated by each agent in the ubiquitous environment.

4 Evaluating the SALSA API

Not much research has been published on evaluating infrastructures (such as middleware, toolkits, or APIs) that support the development of systems. Software Engineering evaluation methods may be adapted to evaluate performance and reliability when producing an application. However, these metrics do not address the end-user experience of software development. They do not provide evidence of the infrastructure's ease of use for implementing systems within a specific domain. Inspired by empirical studies for evaluating infrastructures used for implementing interactive systems [8][4] we identified that for evaluating the ease of use of a middleware, it should be evaluated on how readable the programs that are written with the middleware's programming language are to other programmers, how learnable they are, how convenient they are for expressing certain algorithms, and how comprehensible they are to novice users [8]. In order to evaluate the ease of use of SALSA agents, we conducted two experiments. The first experiment evaluated the API of SALSA in an in-lab experiment, and in the second one, we evaluated the use of SALSA agents as design abstractions for conceiving ambient computing systems. A group of the Object Oriented Analysis and Design (OOAD) class at the CICESE Research Center participated in these experiments.

4.1 Ease of Use of the SALSA API

The objective of this experiment was: "Evaluate the ease of use of the SALSA class framework to develop ubiquitous computing applications." To achieve this objective, an experiment conducted in three separate sessions was carried out.

In the first session, we assessed the participants' experience in developing software systems. Sixteen (16) students participated from the group of nineteen (19) students selected to participate in the experiment. The results from this survey indicated that the participants did not have the same level of knowledge in some issues relevant for the experiment, such as the use of autonomous agents (25% stated that they had no idea or only a vague idea of what an agent is. Their programming and design background was evaluated in order to adapt the experiment to their level of expertise by providing information they would need during the experiment. In the second session we explained to them the concepts of ubiquitous computing, software agents, SALSA API, including a programming example using SALSA, and the in-lab experiment in which they were going to participate. Finally, the evaluation session was conducted in a computing laboratory.

4.1.1 In-lab Evaluation

During this evaluation all (19) nineteen students from the OOAD course participated. Through three programming tasks, participants were asked to implement and extend the functionality of one of the autonomous agents of a ubicomp system that monitors the state of a patient with diabetes, and based on this information, decides on a course

of action. During the experiment, we recorded the participants' comments and questions, and saved their code for further review. When the participants finished their programming exercises, they were asked to answer a questionnaire to evaluate the ease of use of the SALSA API. To illustrate the desired functionality of the system, the following scenario was provided to the participants:

“George, a patient with diabetes, is alone at home watching a soccer game. Suddenly, he feels sick and the ubicomp system detects that he has hypoglycemia. That is, his glucose levels are low (76mg/dl). Thus, the system recommends him to take 2 tablespoons of sugar. A little later, the system measures the glucose level to be lower (70 mg/dl) and starts monitoring the patient’s pulse and his level of perspiration, which at that moment are normal. At the same time, the system notifies George’s daughter of his condition. The system continues perceiving George’s vital signs. As he still presents hypoglycemia, and his pulse and level of perspiration increase, the system decides to send a warning message to George’s doctor”.

The main components of the system depicted in the scenario were modeled with three autonomous agents acting as proxies to the sensors, which monitor the levels of glucose, pulse, and perspiration, respectively. These agents send the information detected to a fourth agent, the patient’s agent, which determines the patient’s health-state and decides what action to execute. Other agents act as proxies to the user interfaces for the patient and doctor. However, we do not elaborate on the design of the system users interfaces, since the aim of this evaluation was to assess the ease of implementing the agent-based system which we identified as the core of ambient computing systems that defines and implements its functionality. Participants were asked to develop the patient’s agent. Finally, participants were given executables of other agents that simulated that they were actually perceiving information from sensors. Thus, the conditions of the patient, such as hypoglycemia (low glucose) or hyperglycemia (high glucose), were simulated, which enabled the participants to verify if their agent acted as expected. We gave the participants the code of the general structure of the patient’s agent. They had to implement the agent’s reasoning and action components as requested in each task according to the SALSA execution model.

Task 1 asked the students to code the conditions for the patient’s agent to diagnose hyperglycemia and the patient’s agent should take an appropriate action. For instance, it had to recommend that the patient drink 2 glasses of water. In task 2, participants were asked to modify the agent to detect hypoglycemia by using the facilities provided by SALSA to derive context. We provided the participants with the code of the XSL filter with the conditions to detect hypoglycemia. The XSL filter read the primary context, which in this case was the information of the levels of glucose, perspiration and pulse, and then deduced a secondary context, it indicated that the patient was having. Finally, in task 3, we required the participants to modify the XSL filter to include the conditions to detect hyperglycemia as was indicated in task 1.

4.1.2 Results from the Inspection of the Code

The source code of the patient’s agent produced by the students was analyzed to evaluate whether they comprehended the execution model and the communication protocol of SALSA agents. Only five (5) participants did not distribute the functionality of the agent as we expected. These participants implemented part or the whole

reasoning logic (a set of rules) in the action component. For this, the information perceived in the reasoning component was passed to the action component to detect the patient's health condition and decide how to act, rather than having the reasoning component do this. Participants faced minimum problems for implementing and understanding the functionality of the perception component. This was due to the fact that most of the agent's perception is left to the SALSA infrastructure which automatically creates and activates this component when the agent is instantiated, and the programmer only has to extract the received information from the launched event. Only three (3) of the participants failed at implementing the perception of information. They did not verify whether the event received in the reasoning component was of type `ArriveSensorDataEvent`, which means that any perceived message, such as a presence message, is analyzed by the reasoning component to find out the patient's condition. Most of the participants were able to successfully implement each of the agent's components, but not without some difficulties. Participants were allowed to check SALSA's user manual during the study or they could ask questions to any of the two graduate students running the experiment. Even though some of the participants understood the execution model of SALSA, they did not remember the name of the abstractions and methods of the SALSA API explained during the 2-hours course. The main concerns of the participants during the first activity were related to XML, since the majority of the participants were but not with XML. During the second and third activity the participants' main questions were about XSL and the facilities offered by SALSA to derive secondary context using an XSL filter, which was new for most of them.

4.1.3 Perception of Ease of Use

When the participants finished their programming tasks, they completed a survey which included topics related with their perception on the use of SALSA. The survey included a questionnaire for evaluating the perceived ease of use of the SALSA API which included six assertions, each one with 7 Likert-scale answers [10]. The participants perceived the API to be easy to use since most of them "slightly agree" (5) or "agree" (6) with all the assertions. The Action component was considered as the agent's functionality that is easiest to implement. A participant wrote that this component was the easiest to implement "because it does not need so many lines of code for implementing the communication of the agent". The agent's functionality that was considered the most difficult to implement, by six (6) of the participants, was to derive context information by using the XSL filter.

5 Evaluating SALSA Agents for Designing Ambient Computing Systems

As mentioned in section 4, the objective of the second evaluation experiment was: "To evaluate the ease of use of SALSA agents in designing a ubicomp system". During this experiment, eighteen (18) graduate students (of the nineteen) participated in a design problem of an ambient computing system by using autonomous agents. For this, the design problem was included in the evaluation exam of the OOAD course taken by these students. The exercise consisted of a description of a use scenario of a

ubicom system for an Airport setting (presented in Fig. 2), a sequence diagram of the interactions among the system's agents, and the following four tasks: a) First, participants had to first analyze the description of the ubicom system and its sequence diagram to identify the main components of the system. b) They were asked to extend the application's functionality elaborating a sequence diagram and a c) component diagram. d) Finally, they had to provide a detailed description of each of the agents' components. Each exercise was reviewed to verify if it fulfilled a set of conditions.

5.1 Results

5.1.1 Exercise a: Create a Diagram Showing the Components of the Original System

The most common mistake was related to conceptual problems of UML for modeling Component Diagrams. For instance, one the majority of the participants (13) made mistakes in establishing the relationships among the components. Two (2) persons described the system's functionality in terms of components or subsystems instead of agents in exercise d. Although most participants said they had a vague idea of what an agent is and of its application, the majority of them (16 persons) had no difficulty identifying the agents of the proposed system as components which wrap the main functionality of the ambient computing system.

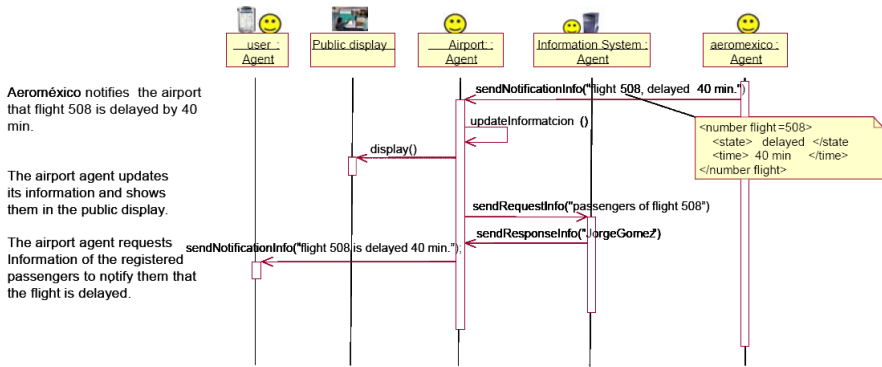


Fig. 2. SALSA Agents interacting for providing flight information to Airport users

5.1.2 Exercise b: Modify the Previous Sequence Diagram, Incorporating the Components that Implement the New Functionality Required by the System

The most frequent mistake was that the scenario was incomplete: Seven (7) participants did not include the interaction with the library service which had to be represented by a proxy-agent; two (2) participants did not specify the interaction to personalize the map for the user's preferences; and one (1) participant did not include the Service Directory, but the User Agent requested the available services to the Information System Agent. In spite of the fact not all of them comprehended the scope of SALSA agents to extend the functionality of a system. This, may be because they

were not familiar with the use of agents for developing systems. The second major error in which five (5) participants incurred was that the Airport Agent was used as a communication intermediary between the User Agent and the Service Directory. In spite of this, the protocol of interaction of SALSA agents was well expressed in the diagram sequence for the majority of the participants. They correctly and clearly specified the agents' interactions to send or request information to other agents. Even though they were not requested to use the real names of SALSA's methods for communication, several of them (8 persons) remembered and used them.

5.1.3 Exercise c: Modify the Components' Diagram Created in Exercise a. to Incorporate the New Components of the Extended System

To check this exercise, we did not take into account the conceptual problems in which participants incurred in solving exercise a. For instance, the participants again made mistakes establishing the relationships between components. Just two (2) participants solve this exercise as we expected. Four (4) of the participants missed specifying a relationship among two of the components, even though in the sequence diagram they clearly specified it. Nine (9) persons were penalized for not including a component: These nine (9) persons did not identify the Library Service as an agent, even though two (2) of them included this agent in the sequence diagram. Two (2) of the nine (9) persons did not include the Directory Service in the component diagram. Some of the mistakes made in this exercise were due to incorrect design decisions made in the sequence diagram. For instance, if the participants did not indicate the interactions with a proxy-agent to the Library in the sequence diagram, they also were not included in the component diagram. However, the other nine (9) participants created agents to add the new functionality to the system following the approach suggested by SALSA to develop ubiquitous computing systems.

5.1.4 Exercise d: Describe the Sequence Diagram you have Extended, Explaining the Behavior of the Agents.

That is, what functionality is implemented by each of the components of the agent (perception, reasoning, action). Five (5) of the participants failed to explain each of the components of the system's agents (perception, reasoning and action), but they gave a general description of each of the system's agent. Most of the participants (14) described the agents' components clearly. They stated how and when the agents perceive; reason based on the perceived information; and specified what events trigger an agent's action. One of them envisioned that the user's agent may perceive from the to-do list of his PDA that he had to buy a book for his daughter.

6 Conclusions and Future Work

The results of the programming exercise provided evidence that the execution model of SALSA and the facilities to implement it are comprehensible. For some of the participants the use of autonomous agents as an abstraction for the deployment of an ubicomp system was not innate since participants had to understand various concepts related with agents. Participants perceived that the SALSA communication protocol was easy to understand. They expressed that for implementing the agent's communication they

just had to select the appropriate methods according to the information that want to be conveyed. The difficulties experienced by the participants for implementing the agents' communication was due to agents using a communication language based on XML, that is semantically richer than that commonly used by object technology. To address this issue, we are proposing ontologies as a means of representing primary as well as the rules that determine secondary context. In this way, ontologies enable programmers to manage the information conveyed by agents. The results of the design exercise shows that students were able to identify the agents as the main system's building blocks from the ubicomp application scenario. These results provided evidence that SALSA is easy to learn and that enables developers to easily conceive an ubicomp system.

References

1. Davies, N., Gellersen, H.-W.: Beyond Prototypes: Challenges in Deploying Ubiquitous Computing Systems. *IEEE Pervasive Computing* 1(1), 26–35 (2002)
2. Kindberg, T., Fox, A.: System Software for Ubiquitous Computing. *IEEE Pervasive Computing* 1(1), 70–81 (2002)
3. Román, M., Hess, C., Cerqueira, R., Ranganatha, A., Campbell, R.H., Nahrstedt, K.: A Middleware Infrastructure for Active Spaces. *IEEE Pervasive Computing* 1(4), 74–83 (2002)
4. Grimm, R.: One world: Experiences with a Pervasive Computing Architecture. *IEEE Pervasive Computing* 3(3), 22–30 (2004)
5. Campo, C.: Service Discovery in Pervasive Multi-Agent Systems. In: *International Joint Conference on Autonomous Agents and Multiagents Systems (AAMAS)*, Italy (2002)
6. Laukkanen, M., Helin, H., Laamanen, H.: Tourists on the Move. In: *International Workshop Series on Cooperative Information Agents (CIA)*., pp. 36–50. Springer, Heidelberg (2002)
7. Villate, Y., Illarramendi, A., Pitoura, E.: Keep your data safe and available while roaming. In: *ACM Proc. of Mobile Networks and Applications, MONET*, pp. 315–328 (2002)
8. Klemmer, S.R., Li, J., Lin, J., Landay, J.A.: Papier-Maché: Toolkit Support for Tangible Input. In: *Proceedings of Human Factors in Computing Systems (CHI)*, Vienna, Austria, pp. 399–406. ACM Press, New York (2004)
9. Rodríguez, M.D., Favela, J.: An Agent Middleware for Ubiquitous Computing in Healthcare. In: Sordo, M., et al. (eds.) *Advanced Computational Intelligence Paradigms in Healthcare–3. Studies in Computational Intelligence*, vol. 107 (to appear, 2008)
10. Davies, F., Bagozzi, R., Warshaw, P.: User Acceptance of Information Technology: A Comparison of Two Theoretical Models. *Management Science* 35(8), 982–1003 (1991)

Defining Re-usable Composite Aspect Patterns: An FDAF Based Approach

Kun Tian, Kendra M.L. Cooper, Kunwu Feng, and Yan Tang

Department of Computer Science
The University of Texas at Dallas
Mail Station ECSS 3.1
2601 North Floyd Road
Richardson, Texas, USA

{kxt056000,kcooper,kxf041000,yxt063000}@utdallas.edu

Abstract. Architecting secure systems is an important and challenging problem. Solutions to model individual, or atomic, security capabilities have been proposed, including security patterns, component based, aspect-oriented, and service-oriented approaches. However, little work is available on how to model reusable compositions of security capabilities, where the security capabilities interact with each other and other parts of the system design. Here, an aspect-oriented approach to modeling composite aspects is presented. The approach is defined as an extension to the Formal Design Analysis Framework (FDAF). The FDAF metamodel is extended to support the static representation of composite aspects and an approach to defining the compositions is introduced. A composite aspect that provides Account Lockout with Selective Event Logging (ALSEL) capabilities is used as an example.

Keywords: Software Architecture, Security, Software Design and Analysis, Formal Design and Analysis Framework, Aspect Reuse, Aspect Composition.

1 Introduction

Architecting secure systems is an important and challenging problem. The importance has been highlighted by the significant cost of security breaches. The average security breach can cost a company between \$90 and \$305 per lost record, according to a new study from Forrester Research [21]. To reduce the risks, it is recognized that security capabilities needs to be pro-actively designed into a system. Some of the challenging issues include how to model and analyze security features in the design, as they can crosscut the design and interact with each other. For example, a web application may require that different data encryption schemes are used based on authenticated users' different privilege levels, in order to optimize the system's performance. The design needs to include interacting authentication, role based access control, and data encryption capabilities.

Existing solutions to modeling security capabilities at the design level include security patterns [2], aspect-oriented approaches [10][11][12], and service-oriented approaches [19][20]. Security patterns are general reusable solutions to commonly occurring security problems in software design. Aspect-oriented design approaches

address the problem of crosscutting concerns, which are elements that are difficult to cleanly modularize in the system [18]. They introduce new modeling elements including aspects, pointcuts, joinpoints, and advice. Aspects are used to model crosscutting concerns separately and are woven into the system. Pointcuts provide the definitions of where and how an aspect will cut into the design; joinpoints are the collection of the actual points of interaction in the design, i.e., where the aspect joins the base design. Advice are design elements that provide functionality when the aspect is invoked. In this way a better modularization and separation of concerns can be achieved than using traditional object oriented design approach.

Using an aspect oriented design approach software designers can integrate a system's security properties into its architecture. Aspects have been used to individually model security design patterns [1][10]. However, when used to model interacting security capabilities, this approach can result in the security aspects crosscutting each other or sharing the same joinpoints in the design. It is argued that a system's consistency can be violated if software designers let aspects crosscut each other, because it risks establishing a circular control structure in which aspects keep invoking each other and as a result the system doesn't halt. Another related problem is known as an aspect mismatch [14][16][3], where it is hard to manage aspect operations regarding to their execution orders and conditions when several aspects share a same joinpoint. It is believed that a flexible approach to address interacting aspects at the design level is needed both to eliminate the inconsistency risks early in the design phase and to resolve the aspect mismatch problem in secure system architecture modeling.

It is argued that to fully specify an aspect oriented system design, its both static and dynamic views shall be needed. The static view is supposed to be used to describe a system's structure and their static relationships with various aspects like where the joinpoints are located and what advices are taken there. The dynamic view is needed to describe the dynamic relationship that may exist between a system's functional components and aspects and those among the aspects themselves. These dynamic relationships may include the execution orders and conditions at a joinpoint where multiple advices are used. By using the dynamic view, potential problems like aspect mismatch and the runtime un-halting structure can be revealed.

To address the above issues, an aspect composition approach for secure system design in static and dynamic views is proposed here. With this new approach software designers and security professionals can produce a composite security aspect by composing several existing security aspects and include it in the repository for reuse. As a result, designers can rapidly model collections of interacting security capabilities without introducing the risks and problems illustrated in the above. The approach is based on the Formal Design Analysis Framework (FDAF) [1]. FDAF is an aspect-oriented architectural framework that supports the systematic modeling and analysis of non-functional properties. In FDAF a non-functional property is represented as a reusable aspect in its repository; a UML extension is defined. A system's semi-formal model can be translated into formal notations for automated analysis. This aspect composition approach could be tailored and applied to alternative aspect-oriented design approaches.

The focus of this work is on the representation of composite aspects, more specifically, the static view of the composite aspects. The dynamic view will be addressed in the next step of the research; this will also consider the checks for consistency between the static and dynamic models. An extension to the FDAF metamodel is proposed and an approach to defining a composite aspect is presented. In the future the formal representations, validation, and analysis of both the static and dynamic views will be investigated. A composite aspect that provides Account Lockout with Selective Event Logging (ALSEL) capabilities is used as an example.

The rest of the paper is organized as the followings. The background for this research is given in Section 2. The FDAF metamodel extension and the approach to defining composite aspects are elaborated using an example in Section 3. The related work is presented in Section 4. The conclusions and future work are in Section 5.

2 Background

FDAF is an aspect-oriented architectural framework for the design and analysis of non-functional properties. The framework has two main parts (Figure 1). The FDAF application development modules support the aspect-oriented design, analysis, and code generation. The FDAF aspect development modules support the definition of a repository of re-usable aspects that provide non-functional capabilities, such as security and performance. The support for composite aspects is the focus of the new extension to the framework.

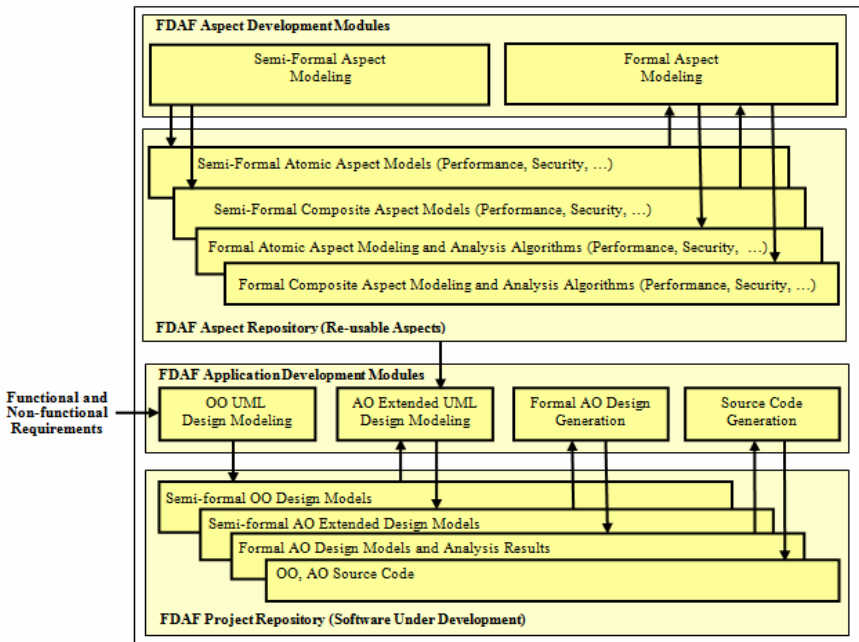


Fig. 1. Overview of the Extended Formal Design Analysis Framework

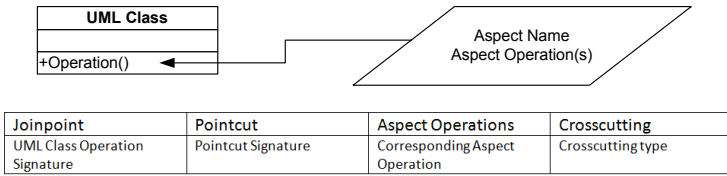


Fig. 2. Aspect Oriented Modeling Extension (Static View)

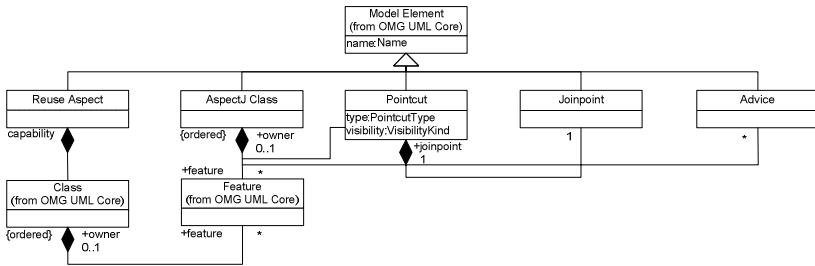


Fig. 3. The Basic View of AOM Core Modeling Capability

The aspects’ metamodel in extended UML is defined in FDAF’s Aspect Oriented Modeling package [1]. It uses a UML extension to assist architects to specify aspects during a system’s architectural design, as illustrated in Figure 2. The parallelogram icon is used to represent the aspect. In the end, the system’s aspect oriented design in UML can be translated into various formal models depending on the types of the aspects. The formal model is then validated using FDAF’s tool support. Software designers can refer to the validation results and decide if the system’s design meets the requirements for the non-functional properties. The designers can go back to reproduce the system’s aspect oriented design until the requirements are met. In addition, code stubs (AspectJ, java) can be automatically generated from the design.

The AOM metamodel for FDAF defines the basic metamodel constructs needed for the development of aspect-oriented models [1]. The basic view of AOM core modeling capability is illustrated in Figure 3.

The reuse aspect in the original AOM package [1] represents a predefined aspect in FDAF aspect repository. Similar to design patterns aiming to provide solutions to frequently occurring problem, reuse aspects provide solutions for realizing a desired functionality that modifies (adds or changes) the behavior of more than architectural design elements, specifically, architectural components. The model element Joinpoint allows architects to identify a well-defined execution point (e.g., operations) of a system in the design. The model element Pointcut allows architects to wrap identified Joinpoints at the design level. It is also the means of representing and identifying Joinpoints. The model element Advice allows architects to express crosscutting rules. An advice is a method-like construct that provides a way to express crosscutting action at the join points that are captured by a Pointcut. The model element AspectJ Class represents an aspect class in an AspectJ program. AspectJClass is a class-like entity that is the basic units for implementing aspect-oriented crosscutting concerns.

3 Introducing Composite Aspects to FDAF

Composite aspect is introduced into FDAF by the proposed approach. A composite aspect is used to model the close interaction of multiple aspects in a specific project's context. The aspects are classified into semi-formal and formal ones. Semi-formal aspects are used to graphically represent an aspect during system design. Formal aspects are used to validate and analyze a system design extended with the corresponding semi-formal aspects (Figure 1).

The current focus of this approach is on semi-formal aspect representation to model interacting aspects (IA) for security concerns. To achieve this goal, the meta-model for FDAF's aspect oriented modeling package (AOM) [1] is extended to introduce the capabilities to model composite aspects.

When a composite aspect is applied to a system's design model in UML, both static and dynamic views are produced, and together they form the aspect oriented system design model. The static view helps the software designer to visualize the interactions between aspects and system components and is also used as an input to formal aspect modeling process. The dynamic view helps clarify the exact sequential relationships between aspect operations and system functionalities, and hence it assists software programmer to implement the aspect oriented solution. To support their rigorous, automated analysis, the semi-formal models of the aspects are translated into formal models. The support for dynamic views, the formal modeling of composite aspects (translation between semi-formal, formal models, and automated analysis) will be the later focus of the new extension to the framework.

3.1 The Extended AOM Metamodel to Support Composite Aspects

The extended AOM metamodel to support composite aspect introduces the following new elements: composite aspect pattern, composite aspect, target joinpoint, base aspect and concurrent aspect (Figure 4).

Composite aspect pattern. This is a particular combination of references to target joinpoints, base aspects and concurrent aspects. It also contains a concern description, design solution description and its rationale to help designers to understand the pattern and hence how to reuse a composite aspect associated with the pattern. The composite aspect pattern is similar in purpose to a traditional design pattern [22].

Composite aspect. This model inherits the name, static view, dynamic view and guidance to use from the reuse aspect in AOM. It is noted that in FDAF the static view of a reuse aspect may also include OCL invariants, pre-conditions, and post-conditions regarding reuse aspect's weaving constraints. A composite aspect is composed of a set of aspects as both base aspects and concurrent aspects indicated by a composite aspect pattern. A composite aspect has full accesses to the functionalities of the aspects used in its composition in order to model their interaction concerns on selected functionalities (operations).

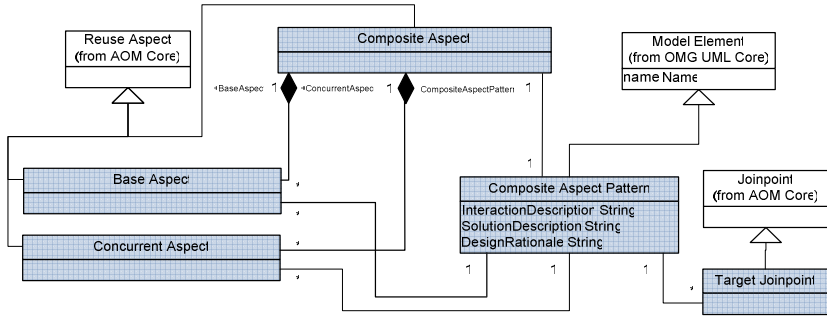


Fig. 4. The Extended FDAF AOM Core Modeling Capability

Base Aspect. This is an aspect whose functionalities (operations) are crosscut by others or takes place first at the target joinpoints for a concern.

Concurrent Aspect. A concurrent aspect is an aspect except the base ones for the same interaction concern. The difference between concurrent aspect and a base aspect lies in that only a base aspect’s operations regarding an interaction concern can be redefined and hence overridden in a composite aspect. Since these operations usually take place first at a shared joinpoint, they form the basis on which the orchestration of the involved aspect operations is performed.

Target joinpoints. This defines a set of shared joinpoints in the system design package or the crosscut functionalities (operations) of the base aspects for the same concern.

The above constructs and the pattern descriptions fully define the interaction concerns for a composite aspect by answering the questions of which aspects are involved, how do they interact with each other and where does the interaction happen.

3.2 The Composite Aspect Modeling Approach

The approach to define a composite aspect consists of three steps. Firstly, an IA must be identified from a project’s security requirements. Secondly, to model the IA a composite security aspect pattern and its semi-formal model are defined from existing security aspects as in FDAF’s aspect repository. Along with the composite aspect definition, several constraints in OCL that specify the IA model may also be produced. Thirdly, software designers use the composite aspect to produce concrete static and dynamic views to capture the IA in the system design.

An Internet Message Access Protocol (IMAP) system’s architecture design [17] is used here to illustrate how to address IAs using extended FDAF. IMAP is an application layer internet protocol that allows a local client to access e-mail on a remote server. The example system implements IMAP Version Four to allow client users to handle emails on its server. Due to the space limit only major steps are illustrated in this section.

3.2.1 Identify the Need for a Composite Aspect Pattern from Interacting Aspects

The need for a composite aspect pattern can be identified at many points in the development. For example, based on their expertise, security experts and designers can define composite aspects they have needed in previous and/or current projects. The key characteristic of an IA is that it involves more than one common security patterns in a specific crosscutting concern. Generally when modeled as a set of separate aspects, an IA may require multiple target joinpoints in the system architecture.

For example, the security policies of the IMAP system demands that: (1) to avoid passwords guessing attacks toward accounts, a protection mechanism against it must be installed in the system; (2) user actions on logins must be logged so that the management can trace user activities and the logs are also kept for auditing and security assessment's purposes; (3) in order to prevent storage volume attack, with which attackers try to turn the system into an inconsistent state by exhausting system storage for logs by issuing large scale passwords guessing attacks, only significant actions such as account lockout events and account management events are logged.

From the requirements of the security policies stipulated in the above, two major security aspects are identified. They are account lockout security aspect (ALSA) and log for audit security aspect (LFA). These two aspects correspond to their respective security patterns [2]. It can be seen that to satisfy the third stipulation of the policies the log for audit security aspect not only crosscuts system functionalities but also crosscuts the account locking operation of account lockout security aspect. Hence, the third stipulation in the security policies is identified as an IA.

3.2.2 Define a Composite Aspect Pattern and Corresponding Composite Aspect

In this step, the composite security aspect pattern capturing the IA and its corresponding aspect definition are produced. To construct the composite aspect pattern, design rationale, (aspects) interaction description and solution description are all manually elicited from the requirements by software designers. The composite aspect pattern for the example system is given in Figure 5.

In the next the composite aspect for the pattern is produced. This process takes six sub-steps. In the first step, its associated composite aspect pattern's base aspects, current aspects and target joinpoints are identified, which is done by reasoning about interactions of involved aspects. In the second step, the corresponding composite security aspect in extended AOM is produced by enclosing the base and concurrent aspects as members. In the third step, the composite aspect's operations and data involved at target Joinpoint are identified. To model the IA, these operations and data are orchestrated into new functionalities (operations) corresponding to its pattern's target joinpoints. In this step, the orchestration of aspect operations takes into consideration the involved operations' action orders, data involved and their interdependencies such as guard conditions. In the fourth step, the orchestration is specified in the composite aspect's OCL constraints. In the fifth step, the composite aspect is validated. The validation of the composite aspect is currently done manually by reasoning about the security requirements and composite aspect patterns. An automatic validation mechanism is currently under development and hence omitted here. Finally the completed composite aspect is then added into the aspect repository as semi-formal composite aspect pattern model.

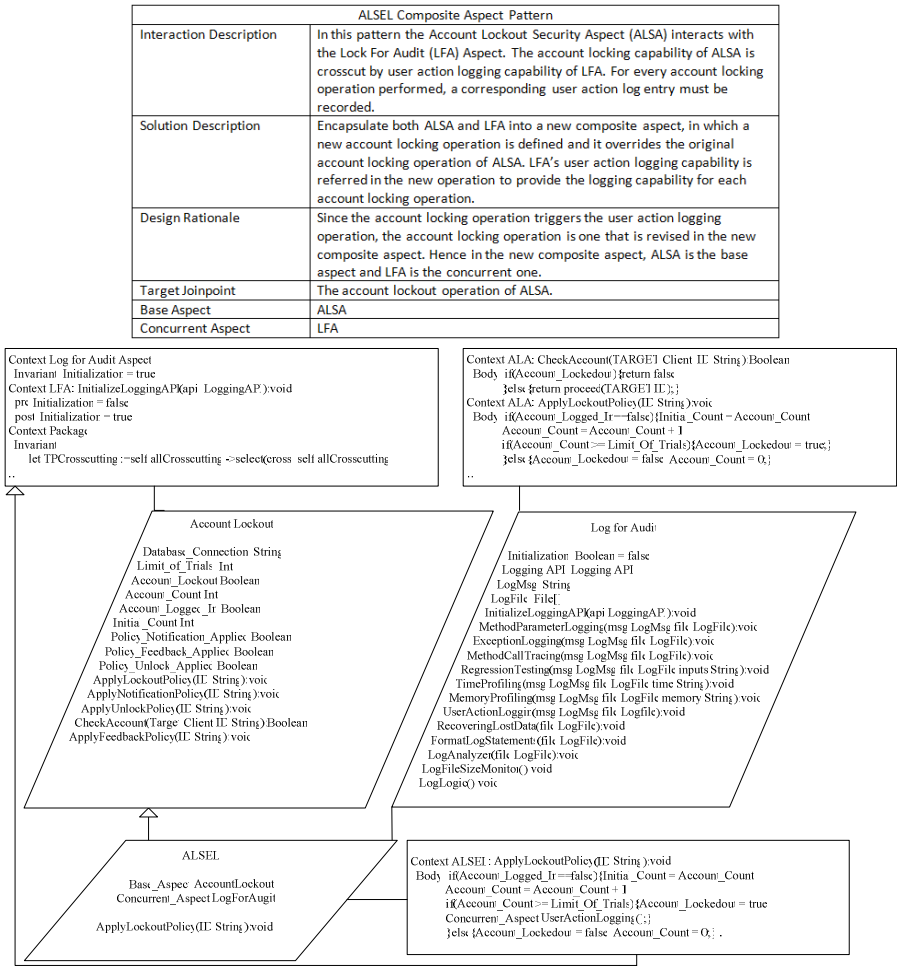


Fig. 5. The Composite Aspect Pattern and its Definition for ALSEL

For the example of the IMAP system design, a composite aspect pattern is defined with ALSA as the base aspect, LFA as the concurrent aspect and the account lockout operation of ALSA as the target joinpoint. The problem is identified as LFA crosscutting ALSA on the target joinpoint. The solution is to orchestrate the involved operations into a new operation in the composite aspect and replace the original account lockout operation of ALSA with the new operation in the system design.

The definition of the composite aspect Account Lockout with Selective Event Logging (ALSEL) in extended AOM is also given in Figure 5. As Figure 5 shows, the new composite aspect ALSEL is produced by enclosing LFA as a concurrent aspect and ALSA as a base aspect. A new operation is defined inside ALSEL to model their interaction on the target joinpoint, which is the original account lockout operation of the ALSA. OCL is used to define the new operation with several constraints. It is

noted that the definition for the new operation is based on both the descriptions in the aspect pattern and the original definition of the target joinpoint, which in this example is the definition of the account lockout operation of ALSA. When applying ALSEL to the system’s design the new operation will replace account lockout operation of ALSA.

3.2.3 Applying the Composite Aspect to a System’s Architectural Design

In this step, the composite aspect’s semi-formal aspect model is applied to the system’s architectural design to produce a static view capturing the system’s static structure and one or more dynamic views capturing the IA. The static view for the example IMAP system is given in Figure 6.

4 Related Work

The application of the aspect-oriented paradigm to software architecture design is of significant interest in the community. Research on aspect identification using architectural reasoning [4], aspect-oriented generative approaches which is focused on Multi Agent System domain [5], aspectual software architecture analysis method [6] and concern modeling [7] have been presented.

A number of aspect-oriented design frameworks have also been proposed. In [8] a formal framework for modeling aspect-oriented system using architecture description language is proposed. Its main purpose is to help the designers to predict the execution of the modeled system so that errors in the architecture design can be removed early. The support for the analysis is similar in concept to FDAF, but the approach does not address the issue of aspect reuse. In [9] a framework for specifying and analyzing aspect-oriented software architecture is proposed. It uses a composition model

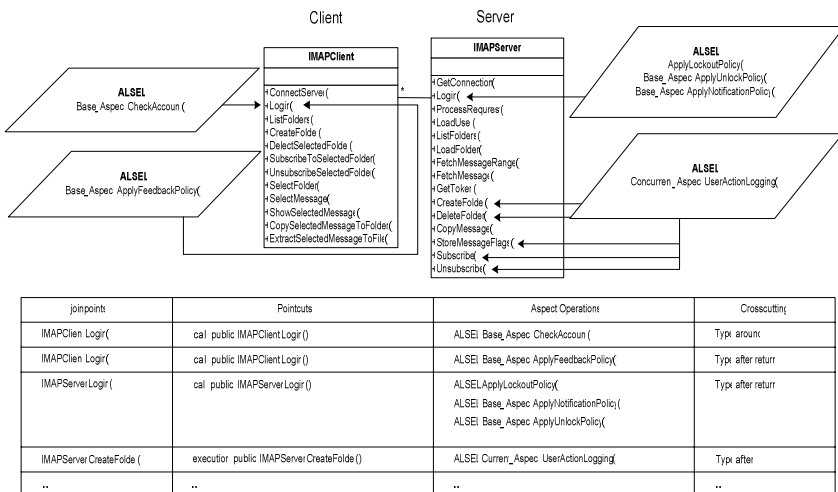


Fig. 6. The Example IMAP System Design with Aspect Composition

and shows the likely transformations to expect when an aspect acts. It also deals with execution issues such as the synchronization of events when an advice happens. This framework does not address the issue of aspect reuse. Unlike the extended FDAF, these aspect oriented frameworks so far have not addressed directly the aspect composition.

Designing secure software systems using the aspect-oriented paradigm has received attention. It is argued in [12] that several observed primary characteristics needed in a secure framework work well with aspect oriented programming model. The authors also developed a tool to address several common security problems in C programs by using these characteristics. In [10], an aspect-oriented design technique for security critical systems is proposed. The basic idea behind this approach is to capture security concerns as aspects and treat them as design patterns. However, this approach neither provides evaluation for different weaving strategies nor gives validation support for the final design. In [11] a more formal approach utilizing Software Architecture Model, Petri nets and temporal logic to define the system and the security aspects is presented. This approach offers a much formal and structured process in applying AOP techniques to design and implementation of security requirements, though there hasn't yet been any tool support.

There is also a continuing interest on composing aspects at their shared joinpoints. In [16] the importance of the "activation order" of aspects that have been superimposed on the same join point is emphasized. In [13] several conflicts related to aspect composition at implementation level are presented. In [14] a detailed analysis of the problem of shared join points is presented. The authors also identified a set of requirements for mechanism of composing aspects at shared join points. Although the above approaches have identified several problems that are related to interacting aspects, they do not provide a systematic support for solving these problems in design level. In [15], a product line development process that structures a set of common and reusable features using aspects is presented. In this process, aspects can be combined into customized components and frameworks to support the product line. In their framework, they propose a dedicated class, called moderator, to manage the execution of aspects at shared join points. However, this approach only considered software product line process, where as the extended FDAF is a more general approach that could be applied to other types of software processes.

5 Conclusions

An aspect composition approach based on FDAF is presented in this work. The FDAF metamodel is extended to support modeling the static view of composite aspects and an approach to defining reusable composite aspects is provided. The composition approach is illustrated that provides Account Lockout with Selective Event Logging (ALSEL) capabilities. The proposed approach helps software designers rapidly integrate their organizations' security policies correctly into the system's architecture without the introducing the risks and problems associated with interacting security capabilities using aspect oriented approach.

There are also some limitations in this approach. Firstly, the number of security aspects in FDAF is still very small. More security aspects capturing common security patterns should be added into FDAF so that its capability to model a broad category of IAs can be improved. Secondly, only the static model is currently supported for the composite aspects. The dynamic models of the composite aspects need to be provided; in addition, consistency checks between the static and the dynamic model need to be defined. Thirdly, the composition of security aspects is performed manually. An automatic aspect composition mechanism is desired to reduce the cost and the possibility of introducing human errors. This will require formal representations of the models. Fourthly, automatic code generation and its reuse for the composite aspect approach in extended FDAF shall be investigated so that this existing useful feature of FDAF will be properly maintained and remain consistent across projects using either normal aspect or composite aspect approach. The future work in this research will be focused on addressing these issues.

References

1. Dai, L.: Formal Design Analysis Framework: An Aspect-Oriented Architectural Framework. Ph.D. thesis, The University of Texas at Dallas (2005)
2. Security Pattern Repository, <http://www.securitypatterns.org>
3. Zhang, J., Li, F., Yang: Aspect-Oriented Requirements Modeling. In: Proceedings of the 31st IEEE Software Engineering Workshop, pp. 35–50 (2007)
4. Bass, L., Klein, M., Northrop, L.: Identifying aspects using Architectural Reasoning. In: Proceedings of Early Aspects 2004: Aspect-Oriented Requirements Engineering and Architecture Design Workshop (2004)
5. Kulesza, U., Garcia, A., Lucena, C.: Generating Aspect-Oriented Agent Architectures. In: Proceedings of the 3rd Workshop on Early Aspects, 3rd Int. Conf. on Aspect-Oriented Software Development (2004)
6. Tekinerdogan, B.: ASAAM: Aspectual Software Architecture Analysis Method. In: Proceedings of WICSA 4th Working IEEE/IFIP Conf. on Software Architecture, pp. 5–14 (2004)
7. Sutton, S., Rouvellou, I.: Modeling of Software Concerns in Cosmos. In: Proceedings of First Aspect-Oriented Software Development Conf., pp. 127–134 (2003)
8. de Paula, V., Batista, T.: Revisiting a Formal Framework for Modeling Aspects in the Design Phase. In: Proceedings of the Early Aspects at ICSE: Workshops in Aspect-Oriented Requirements Engineering and Architecture Design (2007)
9. Sun, W., Dai, Z.: AOSAM: A Formal Framework for Aspect-Oriented Software Architecture Specifications. In: Proceedings of the Int. Conf. on Software Engineering and Applications (2004)
10. Georg, G., France, R., Ray, I.: An Aspect-Based Approach to Modeling Security Concerns. In: Proceedings of Workshop on Critical Systems Development with UML, pp. 107–120 (2002)
11. Yu, H., et al.: Secure Software Architectures Design by Aspect Orientation. In: Proceedings of Tenth Int. Conf. on Engineering of Complex Computer Systems, pp. 45–57 (2005)
12. Shah, V., Hill, F.: An Aspect-Oriented Security Framework: Lessons Learned. In: Proceedings of AOSD Techn. for Application-Level Security, AOSDSEC (2004)

13. Bussard, L., Carver, L., Ernst, E., Jung, M., Robillard, M., Speck, A.: Safe Aspect Composition. In: Proceedings of Workshop on Aspects and Dimensions of Concern at ECOOP 2000 (2000)
14. Nagy, I., Bergmans, L., Aksit, M.: Composing Aspects at Shared Join Points. Technical report, NODe/GSEM 2005: 19–38, University of Twente (2005)
15. Griss, M.L.: Implementing Product-line Features by Composing Aspects. In: Proceedings of the first conf. on Software product lines, pp. 271–288 (2000)
16. Constantinides, C.A., Bader, A., Elrad, T.: An Aspect-Oriented Design Framework for Concurrent Systems. In: Guerraoui, R. (ed.) ECOOP 1999. LNCS, vol. 1628, Springer, Heidelberg (1999)
17. Rhoton, J.: Programmer's Guide to Internet Mail, 1st edn. Digital Press, Boston (2000)
18. Filman, R.E., Elrad, T., Clarke, S., Aksit, M.: Aspect-Oriented Software Development. Addison-Wesley, Reading (2004)
19. OASIS security assertion markup language (SAML) V1.1. Committee specification, <http://www.oasis-open.org>
20. Leune, K., Papazoglou, M., van den Heuvel, W.-J.: Specification and querying of security constraints in the EFSOC framework. In: Proceedings of the 2nd int. conf. on Service oriented computing (2004)
21. Kark, K., Stamp, P., Penn, J., Dill, A.: Calculating the Cost of A Security Breach, Technical report, Forrester Research, Inc. (2007)
22. Beck, K.: Implementation Patterns. Addison-Wesley, Reading (2007)

Implementation Variants of the Singleton Design Pattern

Krzysztof Stencel¹ and Patrycja Węgrzynowicz²

¹ Institute of Informatics, Warsaw University, Poland
stencel@mimuw.edu.pl

² NASK Research and Academic Computer Network, Poland
patrycjaw@nask.pl

Abstract. We present a study of different implementation variants of the Singleton pattern and propose an intuitive definition of this pattern expressed as a first-order logic formula. We also show that our method for automatically detecting design patterns can be used to detect instances of the Singleton with respect to this definition. We also provide data on experiments with a proof-of-concept implementation of this detection method. These experiments prove the efficiency and high accuracy of the method, which is able to detect many non-standard variants of the Singleton in real source code.

1 Introduction

Design patterns [1] facilitate creation of quality designs. As well as being useful during the construction of software systems, they also aid the analysis of existing systems, e.g. reconstructing the documentation of a legacy system from its source code, i.e. *reverse engineering*. The automation of reverse engineering can be more efficient if design patterns are recognized. If they are properly detected, the analysis can appropriately reflect the design intentions. If these intentions are caught in the reconstructed documentation, the future maintenance and development of a legacy system should be far easier and less costly.

The recognition of instances of design patterns in code is difficult, because the patterns are not formally defined. The only formal thing we have is the *canonical form* of each pattern. Instances of design patterns can depart from the canonical form because of specific properties of the chosen programming language and additional design requirements implied by the nature of the solved problem. Furthermore, design patterns are often independently invented and developed by programmers. Although the ideas behind these inventions are similar, details of their implementations can differ. Instances of design patterns are also often tangled together – a particular system function is usually implemented by the cooperation of a number of patterns.

Our aim was to capture the Singleton pattern intent and provide a way to discover as many of its implementation variants as possible. In this paper, we analyse different implementations of the Singleton pattern to find the pattern essence. We also show that our method for automatic pattern recognition can be used to detect variants of the Singleton. We provide the description of a proof-of-concept implementation of our detection method and the results of experimental comparisons with other pattern recognition approaches.

Contributions of this paper

- We present a study of the different implementation variants of the Singleton pattern. The study helps in better understanding of applications of the variants of the Singleton. The list of the variants also forms a benchmark that helps to assess the accuracy of pattern detection tools in their discovery of the Singleton instances.
- We present an intuitive definition of the Singleton pattern that covers different variants of the Singleton identified in the above study.
- We show that our pattern recognition method is able to detect many non-standard implementations of the Singleton pattern (according to the presented intuitive definition).
- We use the proof-of-concept implementation of our methodology to show that it is both efficient and successful in recognizing diverse implementations of Singleton.

2 Motivating Examples

For the last few years, we have observed a continuous improvement in the field of design pattern recognition. Current approaches are capable of detecting a fairly broad range of design patterns, targeting structural as well as behavioural aspects of patterns. However, these approaches are not faultless and sometimes fail to capture source code intent.

The Singleton pattern is the most popular pattern detected by the existing detection approaches. Its canonical implementation is simple, and the intent seems straightforward. However, by carefully analysing the structure of this pattern, we can identify some corner cases among its implementation variants as well as in a usage context.

Firstly, we should realize that approaches solely based on structural relationships are not able to detect the Singleton pattern instances correctly. For example, FUJABA [2], which is one of the state-of-the-art detection tools, utilizes only structural relationships in the recognition of Singleton instances. FUJABA reports a Singleton instance when it finds that a class has a static reference of the Singleton class and has a public-static method with a return type of the Singleton class. Obviously these criteria do not guarantee that a reported class is a true Singleton. FUJABA does not check the usage of a Singleton constructor. FUJABA reports a false positive Singleton in the case shown in Figure 1.

```
public class S1 {
    private static S1 instance = new S1();
    S1() {}
    public static S1 getInstance() { return instance; }
}
public class Usage {
    public useS1() { S1 s1 = new S1(); }
}
```

Fig. 1. Class S1 is not a singleton, but FUJABA reports a false positive

Secondly, it should be understood that the most popular implementation variant with lazy instantiation is *not* the only one available. PINOT [3] goes a step further than FUJABA and addresses the behaviour of the Singleton by searching for a code block representing lazy instantiation. However, PINOT produces a false negative result for the Singleton with an eager instantiation, which is another popular implementation variant of the Singleton pattern.

Thirdly, current approaches focus on searching for a specific construct (structural or behavioural) to be *present* in a code. For the Singleton, it is equally important to verify what is *not present*. A successful detection tool should verify that Singleton instances *do not leak* in an uncontrolled manner. For example, PINOT reports a false positive in the case of Figure 2 because the tool is satisfied to find a static method with a lazy instantiation block and ignores the second static method producing new instances.

```
public class S3 {
    private static S3 instance;
    private S3() {}
    public static S3 getInstance1() {
        if (instance == null) instance = new S3();
        return instance;
    }
    public static S3 getInstance2() { return new S3(); }
}
```

Fig. 2. Class S3 is not a singleton, but PINOT reports a false positive

Additionally, many approaches depend on the access modifiers (e.g. a Singleton constructor marked private). We should remember that the access modifier semantics depend on a programming language, not always guaranteeing a strict access control (e.g. a Java inner class has access to the private attributes of the enclosing class). Furthermore, a Singleton may have subclasses; thus, it may require at least a protected constructor. Moreover, for pattern detection, the requirement of a private Singleton constructor may be too strong. In the course of a development cycle, it may happen that a private modifier of a Singleton constructor is missing but the class still is used in the Singleton manner. For reverse-engineering tools, it is important to capture the code's intent; thus, such a class should be reported as a Singleton candidate. Unlike FUJABA, PINOT requires a constructor to be marked private.

3 Singleton Implementation Variants

In this section, we present different implementation variants of the Singleton pattern. It is worth noting that these variants are *not* disjunctive: they can be combined to form new variants of the Singleton.

Eager Instantiation. In the eager instantiation variant of the Singleton pattern, a singleton instance is created and assigned to a static attribute in an initialization block. Initialization blocks are usually executed at the beginning of a program or when the

class is loaded. Thus, it may happen that the singleton instance is constructed even if it is not accessed. This is a drawback, especially when the construction involves allocation of limited resources. The advantage of this variant is its simplicity and (usually) thread safety (often a language guarantees the thread-safe execution of static initialization blocks).

Lazy Instantiation. The most popular variant of the Singleton implementation involves the lazy instantiation of a singleton instance. An access method checks whether the instance is constructed, and if not, it creates one. This variant avoids the drawback of eager instantiation, as no resources are allocated before the instance is actually accessed. However, this variant must be used carefully in a multi-threaded environment. Without additional synchronization, it may happen that the singleton constructor is called several times, resulting in undesired side effects and resource consumption.

Replaceable Instance. The Singleton intent is to ensure that a class has only one instance. An important question is whether this instance must remain the same throughout the whole execution of a program. The answer is no, an instance can be replaced with some other instance. This feature of the Singleton pattern becomes obvious when we consider a GUI look-and-feel configurator. This is a standard example of the Singleton pattern. It seems natural that a GUI look-and-feel can be changed by a user; thus, its singleton instance must be replaceable. A setter method must exist that allows us to configure a different look-and-feel instance.

Subclassed Singleton. A Singleton class is often perceived as final. It usually has a constructor declared as private, which makes subclassing impossible in most programming languages. However, a singleton class can have subclasses. Subclasses are useful when one wants to change the behaviour of a singleton (one of the advantages of a Singleton implementation over the analogous implementation with all methods being static). The above-mentioned example of a GUI look-and-feel configurator usually involves subclassing of a singleton.

Delegated Construction. A singleton class (or its access method) can delegate the construction of its instance to some other method or class. Thus, it is necessary to analyse the call flow and data flow of a program to identify correctly the instances assigned to the static attributes. A delegated construction may occur when the singleton instance is combined with some other pattern, e.g. the Factory Method, to parameterize the construction of a singleton instance.

Different Placeholder. The canonical implementation of the Singleton pattern requires a static singleton variable to be placed in the singleton class itself. However, there are implementation variants where the static variable is held in a different class. One such is a Java variant in which an inner class is used as a placeholder for a singleton instance. This allows a Java programmer to leverage a language feature to guarantee a correct lazy initialization in a multi-threaded environment.

Different Access Point. It usually happens that a singleton class is an access point to a singleton instance (a static 'get' method). However, it may happen that a singleton instance is managed and accessed via a different class. In this variant, a different class

combines a placeholder function with an access point function. It is a common practice to provide an Abstract Factory implementation that holds the returned instances (products) in the static attributes for future reuse. The implementation of the factory and its products is usually hidden behind the interfaces and separated in a different module. Thus, other parts of a system do not know about the concrete implementations and the singletons.

Limiton. It is worth considering a design concept similar to the Singleton in which the intent is to ensure that a class has a limited number of instances. We call it the *Limiton*. [1] mentions that it is one of the advantages of the Singleton – the possibility of accommodating a policy of a limited number of instances. The similarity between the Singleton and the Limiton is well illustrated by an object-oriented model of a solar system. A single-star system (like the Solar System) might be implemented with a `Star` class having only one instance (the classic example of the Singleton). A binary star system (like Sirius) might be implemented with a `Star` class having only two instances (an example of the Limiton). These two analogous implementations may be considered as two variants of the same design concept. Moreover, such information is worth reporting, because it provides an additional value to an analyst of a reverse-engineered system and describes the design concept behind the programming construct.

4 Singleton Definition

The Singleton’s intent is to ensure that a class has only one instance. It implies that a Singleton instance should be reusable. The only way to keep a Singleton instance for future reuse is to store it in a static (or global) context. Existence of a static (or global) variable of a Singleton type is only a necessary condition. To identify a true Singleton, we must make sure there is no improper usage of a class, i.e. a new instance of a Singleton is instantiated only to initialize a static (or global) Singleton variable. As it might be hard to verify whether for each execution path an instance is assigned to a static variable, we propose only to verify whether such an execution path exists.

More formally, a class C is a candidate Singleton instance if and only if:

- there exists exactly one static attribute A of the type C , and
- for each instance of the type C there is an execution path where the instance is assigned to a static (or global) attribute.

The formula below expresses the above definition using the relations described in Section 5. To cover Limiton candidates as well, we simply replace the “exists exactly one” quantifier with the standard “exists”.

$$\underbrace{\exists! \text{attr. } A : A.\text{staticOrGlobal} = \text{true} \wedge A.\text{type} = C \wedge \exists \text{inst. } I : \text{istype}(I.\text{type}, C) \wedge \forall \text{inst. } J : \text{istype}(J.\text{type}, C) \Rightarrow \exists \text{attr. } B : \text{instance2static}(J, B)}_{\downarrow}$$

The class C is a candidate Singleton.

The above definition does not impose any structural constraint on a program structure except the existence of a static (or global) attribute of a Singleton class. Contrary to other solutions, we do not require the static attribute to be present in a Singleton class itself, nor do we force a static access method to be implemented, nor do we depend on access modifiers.

5 Pattern Detection Tool

In our approach, we use an established program metamodel to formulate the definitions of design patterns in first-order logic. Then we translate the logic formulae to SQL and execute the queries against a database containing a program metamodel. The program metamodel helps to avoid a strong connection to a particular programming language, while the first-order logic definitions of patterns abstract away a programming language or particular analysing technique.

The proposed metamodel consists of the set of core elements and the set of elemental relations among them. The core elements include types, attributes-or-variables, operations, and instances. Most of them have their intuitive object-oriented meaning with one exception – an instance. An instance has been defined as an equivalence class of the relation “objects constructed by the same `new` statement”, i.e. all objects instantiated by the same `new` statement are treated as a single instance. The elemental relations describe the structural (*istype*, *override*) and behavioural (*invocation*, *instantiation*, *input*, *output*, and *instance2static*) features of a program. They model program characteristics such as inheritance trees, call graphs and sets of input and output values, together with possible assignments to variables.

In the Singleton definition, we use only two of the elemental relations: *istype* and *instance2static*. The relation *istype*(*A*,*B*) indicates that *A* is a type of *B* (direct or indirect subtype, equal to). The relation *instance2static*(*I*, *A*) indicates that there is a potential execution path where the instance *I* is assigned to the static (or global) attribute *A*.

Detection of Diverse Design Pattern Variants – D³ (D-cubed) is our tool developed as a proof of concept for our methodology. At present, the tool detects five design patterns (Singleton, Factory Method, Abstract Factory, Builder, Visitor) in Java source code using static analysis techniques and SQL.

The detection process includes the steps parsing, analysis and detection. In the parsing step, we use the Recoder tool [4] to create an abstract syntax tree (AST) from the Java sources, and then we construct the core elements based on the AST. The analysis step involves performing a set of analyses (structural analysis, call flow analysis, and data flow analysis) to discover the elemental relations. During the analysis phase, the transitive closures of relations are computed if necessary. Then, the core elements and relations discovered by the analyses are stored in a relational database. At the detection step, SQL queries are executed to discover pattern instances. The queries follow the design pattern definitions expressed in first-order logic and presented in Section 4.

The set of static analyses used to discover the elemental relations include: structural analysis, call flow analysis, and data flow analysis.

¹ Or its language-dependent equivalent.

Structural Analysis is responsible for discovery of the relations implied by the structure of the program, i.e. the *istype* and *override* relations. First, the direct relations are discovered, and then necessary transitive closures are computed using DFS. *Call Flow Analysis* discovers the relations implied by the structure of invocations, i.e. *invocation* and *instantiation*. It is important that this analysis only focuses on a type hierarchy and method bodies but ignores the data flow; the computed relations might therefore include more elements than the actual number generated during program execution. This feature allows analysis of a broader spectrum of potential program execution paths. Analogously to the structural analysis, the direct relations are first discovered and transitive closures are then computed where necessary.

Data Flow Analysis performs a simple static data flow analysis in a top-down and flow-sensitive way. The analysis discovers the following relations: *input*, *output*, and *instance2static*. First, the operations are sorted using a topological sort. Then, an arbitrary number of iterations is performed to stabilize the input and output sets of each operation and values of each static attribute set as much as possible. In each iteration, the method bodies are interpreted using a simple Java interpreter developed specifically for this purpose. While interpreting, information about possible input, output and static values is collected.

6 Results

We have compared D^3 with the state-of-the-art pattern detection tool PINOT. We have tested the ability of these two tools to detect the Singleton instances in the following sources:

- the demo source of “Applied Java Patterns” [5] (AJP) – an exemplary Java implementation of GoF patterns including the Singleton;
- the Singleton benchmark – our custom set of various implementation variants of the Singleton pattern (see Section 3), available to download at [6]; and
- JHotDraw – a Java GUI framework for technical and structured graphics.

Both D^3 and PINOT successfully reported a Singleton instance in AJP. However, there was a significant difference in the detection of the Singleton pattern between the tools for the Singleton benchmark and JHotDraw.

Table 1 shows the results of the Singleton pattern detection of PINOT and D^3 against the custom Singleton benchmark. PINOT is able to detect only a fairly low proportion of the Singleton variants, whereas D^3 recognizes all of them correctly.

PINOT did not report any Singleton instances in JHotDraw, whereas D^3 recognized five Singleton candidates. After careful analysis, all of them were considered to be true positives. Three candidates are obvious Singleton instances documented in the source code. PINOT did not detect them because one case uses the eager initialization technique and the two other cases represent the Different Access Point variant. The fourth candidate is a more complex Singleton instance because it is the combination of Factory Method and Singleton (the Delegated Construction variant). Finally, the last candidate is a Limiton instance with six instances allowed.

Table 2 shows the running times of the parsing, analysis, and detection phases from the test performed against JHotDraw. The test was performed on a machine with a 2

Table 1. The results of the Singleton pattern detection of D^3 and PINOT against the Singleton benchmark

	PINOT	D^3
Eager instantiation	\times^{FN}	\checkmark^{TP}
Lazy instantiation	\checkmark^{TP}	\checkmark^{TP}
Replaceable instance	\checkmark^{TP}	\checkmark^{TP}
Subclassed singleton	\times^{FN}	\checkmark^{TP}
Delegated construction	\checkmark^{TP}	\checkmark^{TP}
Different placeholder	\times^{FN}	\checkmark^{TP}
Different access point	\times^{FN}	\checkmark^{TP}
Limiton	\times^{FN}	\checkmark^{TP}
Uncontrolled usage (inner class)	\times^{FP}	\checkmark^{TN}
Uncontrolled usage (the same class)	\times^{FP}	\checkmark^{TN}

\checkmark^{TP} the tool correctly reported the pattern instance

\checkmark^{TN} the tool correctly did not report the pattern instance

\times^{FN} the tool did not report the correct pattern instance (false negative)

\times^{FP} the tool reported the incorrect pattern instance (false positive)

Table 2. Detailed running times of D^3 on JHotDraw (45.6 KLOC)

Phase	Time (in seconds)
Parsing	1.29
Structural Analysis	4.01
Call Flow Analysis	2.34
Data Flow Analysis	6.40
Insertion into Database	17.67
Detection	4.13
Total	35.84

GHz Intel Centrino Duo processor with 2 GB RAM running Windows XP and MySQL 5.0.27.

Compared with other tools, D^3 is fairly fast. It is a little slower than PINOT but significantly faster than FUJABA. PINOT analyses JHotDraw in 7 seconds on the same machine, while it takes 20 minutes for FUJABA to analyse Java AWT 1.3.

7 Related Work

Despite a number of developed approaches to design pattern recognition, there is still room for improvement, especially in recognition of behaviour-driven patterns, pattern variants, and performance. Many approaches use only structural information to detect design pattern instances, but there are also several approaches that exploit behavioural information contained in the source code.

Structure-driven approaches are mostly based on type hierarchies, association and aggregation relationships, method modifiers, method signatures, and method delegations. Although these approaches use the same type of information, they leverage various representations and search mechanisms. In addition to structure-driven approaches, some methods targeting behaviour of patterns have been invented. These behaviour-driven approaches employ various types of analysis, including static and dynamic program analysis, fuzzy sets, or machine learning techniques, to capture the intent of code elements.

Some approaches store information in a database (e.g. DP++ [7] or SPOOL [8]). A popular method is to use a logic inference system. [9] utilizes the SOUL logic inference system to detect patterns based on language-specific naming and coding conventions. SPQR [10] employs a logic inference engine; however, to represent a program and pattern definitions it uses a form of denotation semantics known as the ρ -calculus. Another approach is [11], where the design pattern detection method is based on graph similarity. [12] detects design patterns by calculating the normalized cross-correlations of the graphs.

Hedgehog [13] and PINOT [3] both apply static analysis techniques to capture program intent. Hedgehog tries to identify some semantic facts that can be later used by a proof engine. PINOT performs hard-coded static analysis to identify pattern-specific code blocks. [14] and [15] utilize dynamic analysis to understand program behaviour. [14] uses a concept of metapatterns in its search for design patterns. PTIDEJ [16] identifies distorted micro-architectures in object-oriented source code using explanation-based constraint programming. Related work of PTIDEJ, [17], utilizes program metrics and a machine learning algorithm to fingerprint design motifs' roles. Another approach that uses machine learning techniques is [18]. It enhances a pattern-matching system [19,20] by filtering out false positives.

Our tool D^3 utilizes both structural and behavioural information about a program. Its structural model (the core elements and structural relations) follows the set of structural predicates presented in [21]. Compared with other approaches, our method shows high accuracy in detection of the Singleton in real source code. It detects many variants of the patterns. In addition, our prototype implementation shows good performance; this is often a serious issue for methods based on logic programming. Moreover, the proposed method is flexible, allowing us to add a new query or modify existing queries easily. In addition, SQL makes the tool more approachable to an average developer because most developers know SQL (logic programming is not required).

8 Conclusions

We analysed the Singleton design pattern and indicated its diverse implementation variants. Then, we showed a generalized definition of the Singleton pattern that covers all the enumerated variants. We also showed that our detection tool D^3 was able to recognize Singleton instances according to our general definition. Experiments with this tool proved that it detected many non-standard implementation variants of Singleton that are

not recognized by the state-of-the-art tools. Although D^3 considers many more possibilities and recognizes many more constructs, its running time is comparable with other pattern detection tools.

Furthermore, our tool is flexible because the queries used to detect design patterns are stored outside the tool and can be easily modified to repeat design pattern retrieval and to obtain more suitable results.

References

1. Gamma, E., Helm, R., Johnson, R.E., Vlissides, J.M.: Design Patterns. Addison-Wesley, Reading (1994)
2. University of Paderborn, G.: Fujaba., <http://www.fujaba.de/>
3. Shi, N., Olsson, R.A.: Reverse engineering of design patterns from java source code. In: ASE 2006: Proceedings of the 21st IEEE/ACM International Conference on Automated Software Engineering, Washington, DC, USA, pp. 123–134. IEEE Computer Society Press, Los Alamitos (2006)
4. Heuzeroth, D., Aßmann, U., Trifu, M., Kuttruff, V.: The COMPOST, COMPASS, Inject/J and RECODER tool suite for invasive software composition: Invasive composition with COMPASS aspect-oriented connectors. In: Lämmel, R., Saraiva, J., Visser, J. (eds.) GTTSE 2005. LNCS, vol. 4143, pp. 357–377. Springer, Heidelberg (2006)
5. Stelling, S.A., Leeuwen, O.M.V.: Applied Java Patterns. Prentice Hall Professional Technical Reference (2001)
6. Węgrzynowicz, P.: The singleton benchmark., <http://www.dcubed.pl/>
7. Bansiya, J.: Automating design-pattern identification. Dr. Dobbs Journal (1998)
8. Keller, R.K., Schauer, R., Robitaille, S., Pagé, P.: Pattern-based reverse-engineering of design components. In: ICSE 1999. Proceedings of the 21st international conference on Software engineering, pp. 226–235. IEEE Computer Society Press, Los Alamitos (1999)
9. Fabry, J., Mens, T.: Language independent detection of object-oriented design patterns. Computer Languages, Systems and Structures 30(1–2), 21–33 (2004)
10. Smith, J., Stotts, D.: Formalized design pattern detection and software architecture analysis. Technical Report TR05-012, Dept. of Computer Science, University of North Carolina (2005)
11. Tsantalos, N., Chatzigeorgiou, A., Stephanides, G., Halkidis, S.T.: Design pattern detection using similarity scoring. IEEE Trans. Software Eng. 32(11), 896–909 (2006)
12. Dong, J., Sun, Y., Zhao, Y.: Design pattern detection by template matching. In: Wainwright, R.L., Haddad, H. (eds.) SAC, pp. 765–769. ACM, New York (2008)
13. Blewitt, A., Bundy, A., Stark, I.: Automatic verification of design patterns in java. In: ASE 2005: Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering, pp. 224–232. ACM, New York (2005)
14. Hayashi, S., Katada, J., Sakamoto, R., Kobayashi, T., Saeki, M.: Design pattern detection by using meta patterns. IEICE Transactions 91-D(4), 933–944 (2008)
15. Heuzeroth, D., Holl, T., Högrström, G., Löwe, W.: Automatic design pattern detection. In: IWPC, pp. 94–104. IEEE Computer Society, Los Alamitos (2003)
16. Albin-Amiot, H., Cointe, P., Guéhéneuc, Y.G., Jussien, N.: Instantiating and detecting design patterns: Putting bits and pieces together. In: ASE, pp. 166–173. IEEE Computer Society, Los Alamitos (2001)
17. Gueheneuc, Y.G., Sahraoui, H., Zaidi, F.: Fingerprinting design patterns. In: WCRE 2004: Proceedings of the 11th Working Conference on Reverse Engineering, Washington, DC, USA, pp. 172–181. IEEE Computer Society Press, Los Alamitos (2004)

18. Ferenc, R., Beszedes, A., Fulop, L., Lele, J.: Design pattern mining enhanced by machine learning. In: ICSM 2005: Proceedings of the 21st IEEE International Conference on Software Maintenance, Washington, DC, USA, pp. 295–304. IEEE Computer Society Press, Los Alamitos (2005)
19. Balanyi, Z., Ferenc, R.: Mining design patterns from C++ source code. In: ICSM 2003: Proceedings of the International Conference on Software Maintenance, Washington, DC, USA, p. 305. IEEE Computer Society Press, Los Alamitos (2003)
20. Ferenc, R., Gustafsson, J., Müller, L., Paakki, J.: Recognizing design patterns in C++ programs with integration of columbus and maisa. *Acta Cybern.* 15(4), 669–682 (2002)
21. Dong, J., Peng, T., Qiu, Z.: Commutability of design pattern instantiation and integration. In: TASE, pp. 283–292. IEEE Computer Society, Los Alamitos (2007)

Semantic Interactions for Context-Aware and Service-Oriented Architecture

Mehdi Khouja¹, Carlos Juiz¹, Ramon Puigjaner¹, and Farouk Kamoun²

¹ Universitat de les Illes Balears, Palma de Mallorca, Spain
{mehdi.khouja,cjuiz,putxi}@uib.es

² Université de La Manouba, National School of Computer Sciences, La Manouba,
Tunisia
frk.kamoun@planet.tn

Abstract. Designing an architecture that supports pervasive computing usually addresses two issues: the context within interact the system components and the services which are available in the environment. This paper proposes a semantic approach to design an architecture for pervasive systems focusing on these two issues. The ambient is viewed as a set of context elements named CoxEl. Physically a CoxEl is a (mobile) device presents in the environment. The core of a CoxEl is an ontology that represents the context, the services, the resources and the neighbourhood of every CoxEl. This ontology is also used for semantic interaction between the CoxEls. This paper illustrates the different semantic interactions that may occurs among the CoxEls. It also describes the roles that play the CoxEls within the ambient.

1 Introduction

Context-aware systems are heterogeneous systems due to the diversity of devices they incorporate. Due to this, it's necessary to build a system architecture that can deal with all the components. Designing such an architecture has to take into consideration the main characteristics of context-aware systems: the context definition, the services offered and the restrictions of resources. The design process of such a system has to focus on two level of multiplicity: individual level and group level. The first level consists on establishing the structure of one element of the context named CoxEl. This can be done by modelling the context within the CoxEl. In previous work [1], we have chosen an ontology-based model for the CoxEl. The second level of design is to consider the group of CoxEls. The concept of group organises the ambient following certain criteria. In fact, the CoxEls may form groups according to common parameters such as sensing data. Similarity of the service offered within the ambient is also considered a criterion to constitute groups. The CoxEls act like users in virtual social network. Besides, the group concept implies the definition of interactions among the CoxEls. In this paper we present the concept of context group. We also specify a set of interactions related to CoxEl group formation and services. These interactions aim to establish an architecture that is service-oriented for a context-aware system.

The paper is organised as follows: In section 2, the related work about semantic context-aware frameworks is briefly depicted. Section 3 devotes to the CoxEl-based environment. In this section, we introduce the concept of group and describe the CoxEl core ontology. We also introduce the semantic contribution within a layered architecture for context-aware system. Finally, section 4 deals with the semantic interactions that may happen among the CoxEls.

2 Related Works

Various context aware systems have been developed following different architectures. Baldauf et al. [2] made a survey on context-aware systems. The Context Broker Agent (CoBrA) system [3] uses a broker-centric agent architecture to provide services for context-aware systems in smart spaces. It uses an ontology-based context model. Due to its architecture, the CoBrA approach is specific to agent systems. In [4], Gu et al. propose a service oriented context-aware middleware (SOCAM) architecture for the building of context-aware services. They define two-level ontologies for context modeling. The first one is a generalized ontology which defines the basic concept of person, location, computational entity and activity. The second is a domain specific one that defines the details of general concept for a specific application such as a smart home domain. This approach focuses on characterizing user situation with a set of user-defined first order logic rules. In [5], an ontology-based modelling is used for the context. The ontology is person centric. In fact, it describes the context component: devices, tasks, resources, role and preference as related to the person within the context. The architectures studied above use the ontology to describe the entire system and do not focus on the components of the ambient. In our approach, we propose to define the context according to a single element of the context. Then, the union of all the elements will define the system ontology. Since the studied approaches focus on the system before the components, they do not deal with the interactions among the elements. We will focus, in this paper, on describing the set of interactions that may occur among the context elements.

3 Context Element Based Environment

In this section, we describe the context-aware environment. First, a general view of the ambient is described by defining the context. We also introduce the concept of group of CoxEl. Then, the semantic is introduced within an abstract layered architecture. Finally, the core ontology of the CoxEl is explained.

3.1 Environment Overview

To have a good representation of a context-aware environment, it is necessary to define the context. We adopted the definition of context proposed by Dey et al. [6]: *"Context is any information that can be used to characterise the situation*

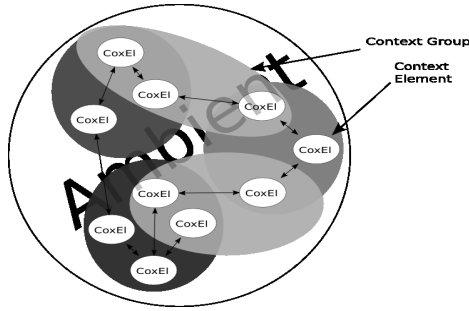


Fig. 1. CoxEl-based Environment Showing Context Groups

of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.”

This definition represents the context as a set of entities. We call them context elements (CoxEls). As shown in figure 1, the CoxEl within the ambient have the capabilities to interact and to form groups according to shared criterion. The group concept is a way to organise the ambient. In fact, CoxEls may form groups depending on their location, type (Sensor, PDA,..), resource restriction and services provided. Like social networks, CoxEls create group based on a common interest. The Group management among CoxEl will be detailed in section 4.1.

3.2 Context Element Layered Architecture

A context-aware pervasive system is composed of three basic subsystem [7]: sensing, thinking and acting. The first one is responsible of collecting context data from the environment. The thinking subsystem includes routines of preprocessing and reasoning. This will produce a modeled and organised context data ready to be stored. The acting subsystem can then perform a specific action on the stored information. The three subsystems can be associated to a layered architecture that incorporates the main functionalities of each subsystem. Figure 2 shows an abstract layered architecture from Baldauf and Dustdar [2] with the corresponding subsystems.

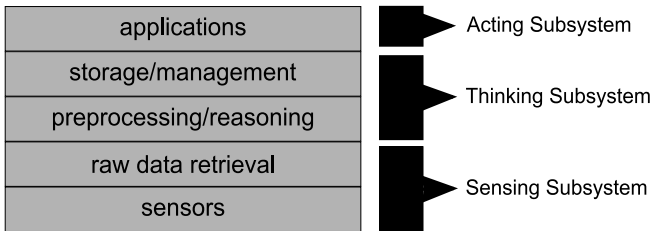


Fig. 2. Abstract Layered Architecture for Context-aware Systems [2]

Our approach to design a context-aware pervasive system consists on adding semantic support to the system architecture. The semantic offers a rich way of modeling and storing context data via ontologies. Thus, the semantic layer will be part of the thinking subsystem. In the next section, the core of the semantic layer is described. In section 4, the interactions between the corresponding semantic layers of CoxEl are discussed.

3.3 Context Element Core Ontology

In previous work [1], we adopted the ontology-based model for context modeling. This is because we have not only to describe the context but also relate the different elements composing the ambient, in order to facilitate the interactions between them. This model constitutes the core of the semantic layer as described in section 3.2.

Our ontology-based model is based on the concept of the context element (CoxEl). The CoxEl is the atomic component of pervasive environment that has awareness and uses resources to perform specified tasks. From this definition, three concepts arise: awareness, task and resource. The awareness concept corresponds to the CoxEl knowledge about itself (self-awareness) and concerning to the ambient around it (surrounding awareness). Since the tasks to be performed are proper to the context element, they are part of the self-awareness. The CoxEl tasks can be of two types: dedicated or acquired tasks. The first type consists on tasks that are integrated in the CoxEl for specific purposes. The second one are tasks learned from other CoxEl. Performing tasks requires various types of

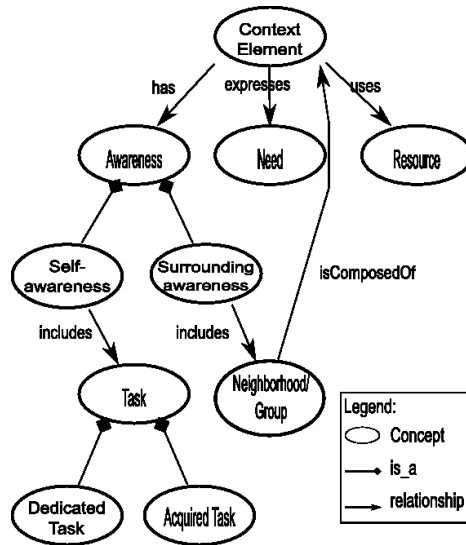


Fig. 3. CoxEl Core Ontology

resources: memory, computing and communication resources. The CoxEl has to adapt itself to the available resources. As the CoxEl can perform tasks, it may need also to access to other tasks from other CoxEl's. This is can be fulfilled through the “need” concept. It includes functionalities for searching services on others CoxEl's. Thus, the CoxEl has to be aware of other CoxEl's in the same context. This is represented by the neighbourhood concept. In fact, CoxEl's that share the same interest or characteristics may form groups. Figure 3 presents the main concepts and relationships of the CoxEl ontology. We present only the concepts proper to our model.

Since the final goal of our research is to design a service oriented semantic architecture, The model described previously integrates in its concepts, the SOA core components. In fact, the “need” concept corresponds to the service requester. Moreover, the “task” concept is the service provider in the SOA. The CoxEl neighbourhood contains the service registry for discovering context services.

4 Semantic Interactions

The interactions that may happen among the CoxEl's are related to the concepts of the core ontology. In fact, we can found group, resource or task related interactions. In this section, group and tasks interactions are described through sequence diagrams.

4.1 Group/Neighbourhood Interactions

The sequence of creating a new group or joining an existing one is shown in Figure 4. The group creation is based on certain criterion such as location, CoxEl type (sensor type), resources or task. The decision of creating or joining a group can be taken automatically by the CoxEl or initiated by the system supervisor. In the first case, the CoxEl has the task to discover periodically its ambient and hence joins to or creates groups. In the second one, the system administrator may change the CoxEl group structure for performance purpose. Within a group, one CoxEl plays a specific role: the group moderator. The main task of the moderator is to maintain the list of CoxEl's that share a common criterion. Since, ambient element have limited resources, they do not have reference to the entire list of CoxEl of the same group. They have only to pinpoint the moderator. The sequence diagram of Figure 4 shows the case where a CoxEl (CoxEl1) starts the group creation process. First, a CoxEl checks whether a group exists according to certain criterion (criterion1). The CoxEl (CoxEl2), receiver of the checking message, searches in its core ontology for the adequate group. If the group exists, CoxEl2 notifies CoxEl1 of the result. It initiates at the same time a join message to the group moderator on behalf of CoxEl2. The Group moderator adds CoxEl1 to group list and notifies it. CoxEl1 creates a new group into its core ontology referencing the group moderator. In the case where the group does not exist, CoxEl1 creates a new group only if CoxEl2 shares the same criterion as CoxEl1.

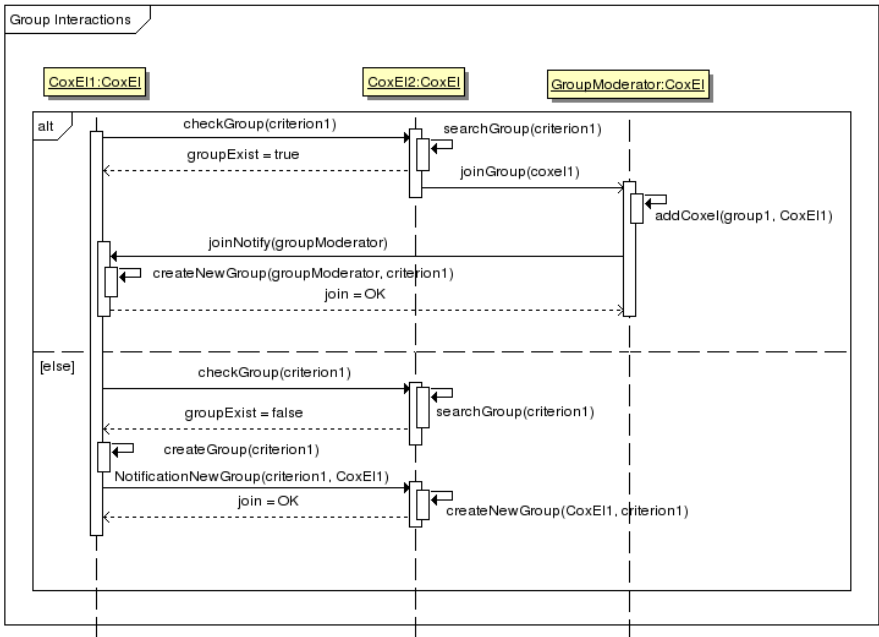


Fig. 4. Sequence Diagram for Group Interactions

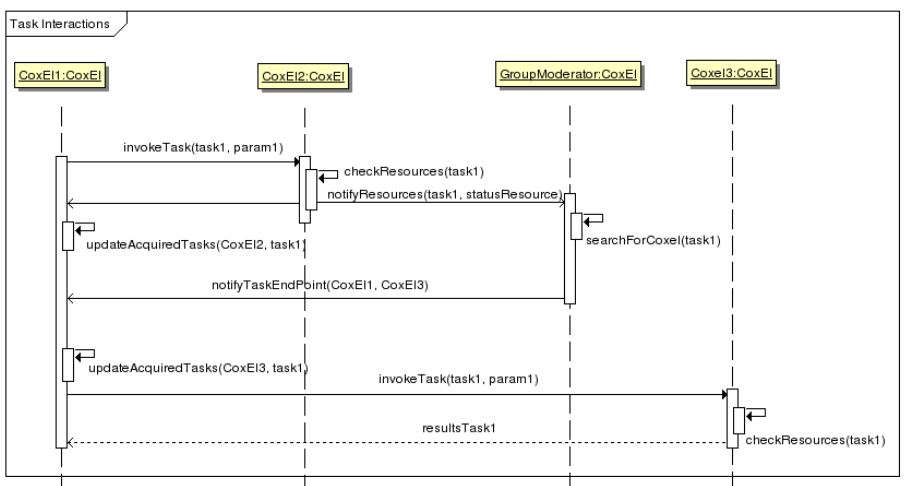


Fig. 5. Sequence Diagram for Task Interactions

Then, the group creator attributes itself the role of moderator unless resource restrictions. It notifies CoxEl2 of the new group. The later CoxEl update its core ontology by adding a new group.

4.2 Task Interactions

The sequence diagram depicted in Figure 5 shows the interactions for task invoking. First, CoxE11 invokes a specific service to CoxE12 by passing the corresponding parameters. Then, CoxE12 checks the resources needed to perform this service. A resource failure forces CoxE12 to contact with the group moderator corresponding to the service criterion in order to look for another CoxEl. The Moderator looks for a CoxEl that can perform the service invoked by CoxE11. Once it is found, it sends a message to the invoker with the new end point information. At this moment, CoxE11 has to re-invoke the desired service to the new CoxEl (CoxE13). Before performing the task, CoxE13 has to check its resources. Once the service executed, the results are send back to CoxE11.

5 Conclusion and Future Works

In this paper we presented a set of semantic interactions within a context-aware system. The Context elements (CoxEls) use a core ontology to perform interactions. These interactions permit to the CoxEls establishing groups within the context and offering tasks. The concept of group is a way to organise the context information within the ambient. In fact, the CoxEls form groups depending on criterion such as sensing information, resource availability or common tasks. The task interactions give the system architecture a service-oriented characteristic. The panoply of interaction can be extended to the resources. Designing resource-related interactions is one of our future work. Since, we described the interaction via an UML unformal approach (sequence diagram), we will consider a design with an interaction specific language such as ISDL (Interaction System Design Language). We have also to determine whether ISDL supports semantic interactions. An implementation of the designed interaction through a real case of study will be done in the future.

Acknowledgement

This work is partially supported by the project *TIN2007-60440 Arquitectura Semántica Orientada a Servicios (SOSA)* under *Programa Nacional de Tecnologías Informáticas* of Ministerio de Ciencia e Innovación, Spain.

References

1. Khouja, M., Juiz, C., Lera, I., Puigjaner, R., Kamoun, F.: An ontology-based model for a context-aware service oriented architecture. In: SERP 2007, vol. 2, pp. 608–612. CSREA Press (June 2007)
2. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. *Int. J. Ad Hoc Ubiquitous Comput.* 2(4), 263–277 (2007)
3. Chen, H., Finin, T., Joshi, A.: An intelligent broker for context-aware systems. In: *Adjunct Proceedings of Ubicomp 2003*, pp. 183–184 (October 2003)

4. Gu, T., Pung, H.K., Zhang, D.Q.: A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications* 28(1), 1–18 (2005)
5. Moore, P., Hu, B., Wan, J.: Smart-Context: A Context Ontology for Pervasive Mobile Computing. *The Computer Journal* (2008); bxm104
6. Dey, A.K.: Understanding and Using Context. *Personal Ubiquitous Comput.* 5(1), 4–7 (2001)
7. Loke, S.: *Context-Aware Pervasive Systems: Architectures for a New Breed of Applications*. Auerbach Publication (2006)

ReWiSe: A New Component Model for Lightweight Software Reconfiguration in Wireless Sensor Networks

Amirhosein Taherkordi, Frank Eliassen, Romain Rouvoy, and Quan Le-Trung

University of Oslo, Department of Informatics
P.O. Box 1080 Blindern, N-0314 Oslo
{amirhost, frank, rouvoy, quanle}@ifi.uio.no

Abstract. Wireless Sensor Networks (WSNs) are increasingly being deployed for applications with dynamic requirements. Moreover, these applications are likely to be run on nodes with different sensing parameters and capabilities. Addressing such issues in the middleware layer and application layer, beside the consideration in the lower layers, is of high importance. *Reconfiguration* of application software has been identified as an effective approach for handling dynamicity issues. However, the special characteristics and limitations of WSNs make the requirements to the reconfiguration mechanism quite different from what has been previously proposed for other types of platforms. In this paper, we propose a new software component model, named ReWiSe, for achieving *lightweight* and *fine-grained* software reconfiguration in WSNs. In this model, a component can be reconfigured at the behavior-level instead of at the component-level. We discuss how the new component model can make the reconfiguration process lightweight in terms of component state preservation, component dependency checking, and new update unit granularity.

Keywords: Component Model, Wireless Sensor Networks, Reconfiguration, Adaptation.

1 Introduction

Deployments of WSNs have increased considerably over the past few years. There are more applications running on WSN platform, and more technical issues on different aspects arise. Challenges become more significant when WSNs are exploited in the new emerging applications not limiting themselves to a single function called “sense and send” with trivial local data processing tasks [1], [2]. Applications for WSNs are gradually moving towards ubiquitous computing environments, where a high number of computing devices deal with dynamic requirement and unpredictable future events [3]. In such an environment, in addition to the basic tasks, an application needs to adapt its behaviors and functionalities to cope with changing environmental conditions, and with different capabilities of each individual sensor node in the network. As different

nodes are expected to run different tasks, software with adaptable functionalities becomes an essential need. Moreover, application in which a high number of nodes are deployed in inaccessible places, individual software updating becomes an impractical solution.

One of the popular mechanisms for handling application dynamicity is the provision of a set of generic middleware services performing reconfiguration tasks [4]. The main advantage of this technique is that the reconfiguration mechanism is kept separated from the application logic. The performance of the middleware proposal for dynamic applications depends on two major factors. Firstly, we should examine to what extent a typical application module is reconfigurable. In some cases, due to the tight coupling between the modules reconfiguring a module needs to update some other parts of application, while in a highly reconfigurable model the update is limited to the part that really needs to be updated. Secondly, the mechanisms by which a module is reconfigured effects the performance of such middleware. In this paper, we focus on the first factor and propose a model as the building block of reconfigurable WSN application in order to improve the performance of the reconfiguration process.

The techniques for application module development in the WSNs are highly dependent on the kind of software unit provided by the underlying operating system. Since the most popular operating systems for sensor nodes such as TinyOS [5] or Contiki [6] are based on *software component models*, contributions to the application and middleware are mostly inspired by such models. To generalize the proposal of reconfigurable application module, we adopt the software component model as unit of development in the WSNs and describe our new model.

Beside the earlier component models not supporting dynamic reconfiguration of components [7, 8, 9], some recent component models are enhanced with features enabling reconfiguration of software component [10, 11]. Basically, these models are targeting large-scale systems with sophisticated models of component integration and high amount of computing resources.

In this paper, we propose a new component model which is more suitable for performing lightweight reconfiguration and adaptation in sensor applications. Using this model, the reconfiguration is limited to the part of a component that really needs to be updated, rather than replacing the whole component with the new one. New concepts in the proposed model simplify the general steps of a typical reconfiguration model. By exploiting this model in designing WSN application software, the reconfiguration program performs its task in a lightweight manner. The notion of *behavior reconfiguration* makes it possible to upload only the changed part of software to the nodes, instead of uploading the whole software component. We believe that this model can be also exploited in other application areas such as mobile applications and embedded systems.

The rest of this paper is organized as follows. The next section introduces the concept of reconfigurability. In Section 3, we describe the ReWiSe component model and its constituents. Then, in Section 4 we discuss how a ReWiSe-based system can be reconfigured in a lightweight manner. Related work is presented in Section 5. Finally, Section 6 concludes the paper and discusses future work.

2 Reconfigurability

Dynamic reconfiguration of software components has been a popular approach to meet the dynamic requirements of applications. Dynamic reconfiguration might include stopping and starting of applications and their components, configuring their connections and parameters, or changing device settings [12]. A reconfiguration service is responsible for safely reconfiguring from the currently running application variant to a new application variant selected by a professional user of the system or existing adaptation mechanism. In the context of WSNs, reconfiguration become fully effective if the reconfiguration service retains fine-grained control over what is being reconfigured, by updating only some selected functionalities to minimize energy consumption. In other word, the ideal way of WSN application reconfiguration is to limit updates to only the portion of software which really needs to be updated rather than updating the full application image. However, in some sensor platforms, for software updates to be successful, the whole system is required to be restarted after performing reconfiguration.

Basically, reconfiguration mechanisms are proposed based on the type of application building blocks. For a component-based framework, two main mechanisms could be employed, namely, *parameter* and *component* [12], [13]. In the former, a component is reconfigured by updating its configuration parameters. This mechanism supports fine tuning of applications through the modification of the values of component configuration parameters and deployment parameters. In fact, in this method the behavior of a component is controlled by manipulating the component or application configuration parameters. As an example, the component collecting sample data from a sensor can be reconfigured through updating only the value of the `SampleRate` attribute (the frequency at which sensors send messages). In employing component reconfiguration mechanism, the whole component can be reconfigured. In particular, component-scope reconfiguration allows the modification of service implementation (replacement of a component), adding a new component, and removing running components. Parameter-scope is an effective way to implement variability, but it is less powerful than component adaptation.

In our model, the dynamic reconfiguration mechanisms are described based on software architecture elements known as *components*. The component model defines constructs for structuring the code on the sensor nodes. Basically, every component represents a single unit of functionality and deployment. As illustrated in Figure 1, every component can interact with its outside through *property*, *interface*, and *receptacle*. An interface specifies a set of operations provided by a component to others, while receptacles specify the set of interfaces a

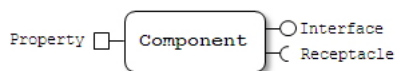


Fig. 1. A typical component model

component requires from others. Therefore we are able to reconfigure the system by switching from an old component to a new one implementing the same interfaces.

3 ReWiSe

As mentioned in the previous section, component-scope reconfiguration is proposed for manipulating the whole component. In general, four forms of component reconfiguration can be envisaged: adding a new component, removing an existing component, replacing a component with a new one and migrating a component. Depending on the form of reconfiguration, system consistency should be preserved during reconfiguration time. For instance, in the component replacement form, the currently running component might be in interaction with others, thereby the reconfiguration mechanism should check somehow the dependencies of the replacement candidate to other components. Likewise, the state of a new component should be updated with the last state of the old component. Therefore, for each form of configuration, several checking tasks should be carried out in order to keep system integrity and consistency at an optimum level.

Let us consider the sample component configuration depicted in Figure 2. Suppose the `Sampler` component is the candidate for replacement with a new component. We also assume that the difference between the old and new version is in the implementation of the `IReport` interface. The four main tasks in carrying out the reconfiguration are as following: (i) checking that the component is not being in interaction with the `Logger` component before starting reconfiguration, (ii) checking that the component is not being in interaction with the `Publisher` component before starting reconfiguration, (iii) storing the state of the component, and (iv) creating the new component and transferring the last state to it. Moreover, there are some other small tasks that should be considered for executing a perfect reconfiguration. Although it seems that these steps are necessary in general regardless of the underlying platform limitations, we believe that it is possible to skip some of these steps in order to have a light-weight fashion of reconfiguration. Especially, in the context of WSNs, existing limitations such as resource usage and network bandwidth motivate us to have a more fine-grained approach to the component construct and its interaction model. ReWiSe is a novel definition of software component that is empowered with new features for having a more dynamic component model, along with the basic functionalities of a typical component model.

The question that may arise from the above sample is: “for updating an interface implementation in the `Sampler` component why do we need to replace the whole current component with a new one?” To clarify, it might be better to update only the part of the component that really needs to be updated rather than replacing it with a new one. ReWiSe aims to achieve such a reconfiguration mechanism.

Figure 3 illustrates the constituents of the ReWiSe component model. Like other popular component models, the interaction of ReWiSe with other

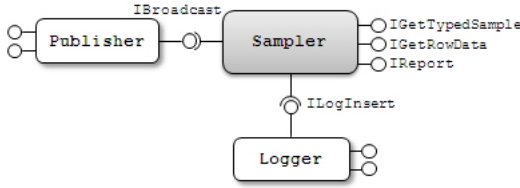


Fig. 2. Sample component configuration

components is established through interfaces, receptacles, events, listeners and properties. In fact, the outer white box is similar to what has been described in the previous models [15]. The main dissimilarity is in the implementation of component interface. Particularly, in this model an interface is not implemented as a `method` within the component body, but it is implemented in a separate component containing “just” the implementation of that interface, and no more functionalities. Let us call these components *TinyComponent*. Therefore, for each interface of the *main component*, we have a corresponding *TinyComponent* implementing that interface. The main component is wrapped with *Interface Interceptor Wrapper* to route the interface calls of other components to the corresponding *TinyComponent* of an interface.

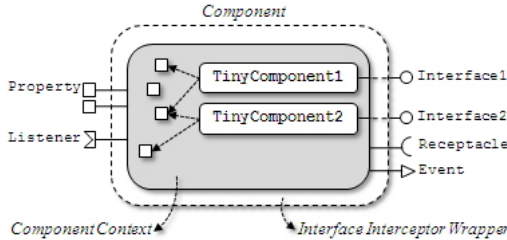


Fig. 3. ReWiSe Component Model

Figure 4 shows the template for declaration of a ReWiSe component. In the top box, the main component is described and two next boxes illustrate the implementation of *TinyComponents*. In the following sections, we explain the concepts of ReWiSe in detail according to the template illustrated in Figure 4.

3.1 TinyComponent

TinyComponent is the representative of an interface implementation. For each interface of the main component, there is a corresponding *TinyComponent* containing just one `method` that realizes the interface. For example, in the bottom of Figure 4 there are two *TinyComponents* implementing the two interfaces of *MainComponent*. New issues arise related to interaction between *TinyComponent* and other elements inside the main component, while its connection with the out-


```

MainComponent implements interfaced, interface2 {
    property1;
    property2;
    interfaced(params1){
        Wrapper.interfaced(params1);
    }
    interface2(params2){
        Wrapper.interface2(params2);
    }
    ...
}

Wrapper{
    interfacedImpl = null;
    interface2Impl = null;
    interfaced(params1){
        if (interfacedImpl == null){
            read the mapping file;
            find and load the corresponding TinyComponent;
            interfacedImpl = TinyComponent1;
        }
        // safe state checking - part1
        interfacedImpl.method(params1, this);
        // safe state checking - part2
    }
    ...
}

TinyComponent1 implements interfaced {
    method(params1, mainComponentRef){
        //interfaced implementation
    }
}

TinyComponent2 implements interface2 {
    method(params2, mainComponentRef){
        //interface2 implementation
    }
}

```

Fig. 4. Declaration of a Component in the ReWiSe model

side world of the main component is in the same fashion as in previous models. Firstly, unlike direct interface implementation in the main component which has easy access to the main component variables, in the ReWiSe model `TinyComponent` is located outside the scope of the main component, so a `TinyComponent` is not able to reach to the variables of the main component. This problem is resolved by passing a reference of the main component to the `TinyComponent`, thereby the scope is reachable for `TinyComponent` through a variable containing a pointer to the main component (see Figure 4). Second question that arises is how `TinyComponents` can interact among themselves, like what occurs between methods in common component models. Like the first problem, this case is originated from the fact that the scope of the main component is not accessible from the `TinyComponent`. Similarly to the resolution of the first question, the reference of the main component passed to the `TinyComponent` can give access to every thing inside the main component, namely, variables and interfaces. In the next section, we discuss in more details how *Interface Interceptor Wrapper* facilitates such a calling among `TinyComponents`.

3.2 Interface Interceptor Wrapper

To systematically access `TinyComponents`, ReWiSe should be enhanced with a mechanism capable to route calls for a specific interface to the corresponding `TinyComponent`. Since `TinyComponents` are going to become the new candidate for replacement in the forthcoming reconfiguration mechanism, component can not maintain the name of a `TinyComponent` in a hardcode manner. Therefore, the interface interceptor wrapper is responsible for handling dynamically the references to `TinyComponents`. The first time a service is executed, the wrapper reads from its *mapping configuration file* the name of the `TinyComponent` implementing the service and then caches the reference to the corresponding `TinyComponent` in its local data. Afterward, other requests for that service will be automatically forwarded to the assigned `TinyComponent` (see `Wrapper` block in Figure 4). The configuration file contains a set of structured data indentifying the name of the corresponding `TinyComponent` for each interface. Moreover, the wrapper is responsible for passing the reference of the main component to the `TinyComponent` (“`this`” in Figure 4).

3.3 Component Context

One of major challenges in reconfiguration mechanisms is how to preserve the state of a component during the reconfiguration period. Since the unit of replacement in ReWiSe is the stateless `TinyComponent`, the replacement candidate has not any state to miss. In fact, the state of the component is preserved in the main component. *Component Context* is an abstract concept indicating the current values of all private and public member variables in the main component. The context is accessible for `TinyComponent` through the main component reference passed to the `TinyComponent` method.

4 ReWiSe-Based Reconfiguration

Let us refer again to the sample component configuration shown in Figure 2, and consider how the specification of ReWiSe modifies the reconfiguration mechanism. Figure 5 illustrates the new situation, where `Sampler` is redesigned according to the standards of the ReWiSe model. Therefore, the three interfaces exposed from `Sampler` should be implemented in three separate `TinyComponents`.

The reconfiguration logic is still the same as the previous one: updating the implementation of `IReport` interface. To this end, at the first step the interaction of `Sampler` with the other components should be checked to make sure that the component is in a safe state for removal. As illustrated in Figure 5, in this case only the dependencies of `TinyIReport` to others should be checked instead of checking the dependencies of the whole main component to others (later in this section the checking mechanism is explained). Upon transition to a safe state for reconfiguration, rationally the state of `Sampler` should be considered. Since in the new model the state is captured by `Sampler`, not by the replaceable `TinyComponents`, we do not need to consider the step of state preservation.

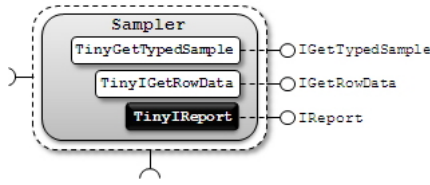


Fig. 5. Sample component configuration based on ReWiSe model

Basically, the three main design concepts of ReWiSe keep the component in a high degree of reconfigurability. Firstly, the notion of TinyComponent makes the behavioral-level reconfiguration possible; thereby if a portion of component (interface implementation) needs to be repaired we do not need to change the full component image. The other advantage of using TinyComponent is that the safe state can be determined only by checking the interactions of the candidate TinyComponent with others rather than checking the interactions of the whole main component. Secondly, the interface interceptor wrapper undertakes the action of switching from the old TinyComponent to the new one. As mentioned before, a mapping configuration file is attached to the wrapper to specify the map between each interface and its corresponding TinyComponent. The final improvement is offered by component state management mechanism. In fact, in previous component models for WSNs, the *stateful* main component is subject to replacement, while in ReWiSe the *stateless* TinyComponent becomes the new replaceable unit. Consequently, component variables maintain their values (component context) during the component lifetime.

In this paper, we discussed the reconfiguration of TinyComponents, whereas the main component containing TinyComponents may need to be reconfigured as a whole. Since ReWiSe follows the standards of popular component models, the available reconfiguration mechanisms for such models are applicable reconfiguring a complete ReWiSe component as well.

4.1 Safe State for Reconfiguration

The safe state for performing reconfiguration is defined as a situation in which all instances of a component are temporarily idle or not processing any request. That is, in a typical component model the interactions among components must be monitored in order to determine when the target component reaches a configurable state. In ReWiSe, the interface interceptor wrapper is enhanced with a *farFromSafety* variable for each exposed interface. This variable will be updated in the application scope each time that the interface is called. Before calling a service, *farFromSafety* is increased by one and after running the service, this variable is decreased by one. Thus, at the arbitrary time of application execution, the value of *farFromSafety* for a specific interface indicates the number of TinyComponent instances running in the memory. In the template declared in Figure 4, the safe state checking code appears in the **Wrapper** block, before and after calling the method of TinyComponent. Once this value reaches zero,

all other requests for that interface will be blocked and the reconfiguration unit loads the new TinyComponent. Afterwards, the blocked requests are served by the new TinyComponent in stead.

4.2 A Typical ReWiSe-Based Application

We are currently working on a *home monitoring application*. Home monitoring systems have been in use for many years [17]. The emergence of wireless technologies reduces the cost of employing such application through easy deployment of monitoring devices. Future home monitoring applications are characterized as being filled with different sensor types that can determine various types of context information and react to them through actuators, e.g., if the temperature sensors in a room report unusual data for a time period the smoke detector sensor should react and increase the sensing sample rate. The various dynamic scenarios in such applications make the application design method different from others in terms of application reconfigurability and adaptability. Our hypothesis is that by using ReWiSe as the building block of such application the cost of dynamic reconfiguration will be reduced.

5 Related Work

The early well-known component models in distributed system area do not support inherently dynamic reconfiguration of components [7, 8, 9], because these models provide the basic building blocks for component-based software, and the core design aspects of such models do not consider features such as reconfigurability of component.

In the context of reconfigurable component models, Fractal has been known as a pioneering model for dynamic software applications [10]. Fractal can be used with various programming languages to design, implement, deploy and reconfigure various systems and applications, from operating systems to middleware platforms. Although Fractal is a comprehensive and extensible model for component composition and assembly, its minimal core is a heavyweight extensible unit with various features suitable for large scale applications needing different degrees of reconfiguration for different software granularity. Moreover, concepts in ReWiSe and Fractal are different to some extent. For instance, the content part of Fractal is devoted for handling component compositions, while in ReWiSe we adopt component context to preserve component state. Also, the interactions among TinyComponents in ReWiSe are handled more easily in compare to the model of interactions between sub-components in Fractal.

A few works have been done recently to enable high-level software reconfiguration for WSN applications [14, 15, 16]. All these proposals adopt the general common component model as the basic building blocks for application, regardless of the complexity of reconfiguration in terms of space, computation and communication.

6 Conclusions and Future Work

This paper presented a new component model, called ReWiSe, to facilitate dynamic reconfiguration of applications in resource-limited networks such as WSNs. The component model introduces the notion of TinyComponent for achieving a lightweight behavior-level reconfiguration for WSNs. Using ReWiSe, not only a consistent reconfiguration is guaranteed, but also the reconfiguration overhead is reduced by updating only the portion of a component (TinyComponent) that really needs to be updated.

We are currently developing a middleware to enable reconfiguration and adaptation for dynamic WSN applications. The performance of the middleware is expected to be better when the application building blocks are ReWiSe components. One possible future work is to exploit ReWiSe in other computing areas such as mobile or embedded systems.

Acknowledgments. This work was partly funded by the Research Council of Norway through the project SWISNET, grant number 176151.

References

1. Puccinelli, D., Haenggi, M.: Wireless Sensor Networks: Applications and Challenges of Ubiquitous Sensing. *IEEE Circuits and Systems Magazine* 5(3), 19–31 (2005)
2. Costa, P., et al.: The RUNES middleware for networked embedded systems and its application in a disaster management scenario. In: *Proc. of the 5th Int. Conf. on Pervasive Communications (PERCOM)* (2007)
3. Akyildiz, I.F., Kasimoglu, I.H.: Wireless Sensor and Actor Networks: Research challenges. *Ad Hoc Networks Journal* 2(4), 351–367 (2004)
4. Alia, M., Hallsteinsen, S., Paspallis, N., Eliassen, F.: Managing Distributed Adaptation of Mobile Applications, In: *Proc. of the 7th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS)* (2007)
5. Hill, J., et al.: System Architecture Directions for Networked Sensors. In: *Proc. of International Conference on Architectural Support for Programming Languages and Operating systems (ASPLOS)* (2000)
6. Dunkels, A., Grönvall, B., Voigt, T.: Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors. In: *Proc. of the First IEEE Workshop on Embedded Networked Sensors* (2004)
7. Microsoft, COM, <http://www.microsoft.com/com>
8. Sun Microsystems. EJB, <http://java.sun.com/products/ejb/index.html>
9. OMG. CORBA, Object Management Group, <http://www.omg.org>
10. Bruneton, E., Coupaye, T., Leclercq, M., Quma, V., Stefani, J.B.: The FRACTAL component model and its support in Java. *Softw., Pract. Exper.* 36(11-12), 1257–1284 (2006), <http://fractal.objectweb.org>
11. Coulson, G., et al.: A generic component model for building systems software. *ACM Trans. Computer Systems*, 1–42 (2008)
12. McKinley, P.: Composing Adaptive Software. *Computer* 37(7) (2004)
13. Poladian, V., Sousa, J.P., Garlan, D., Shaw, M.: Dynamic Configuration of Resource-Aware Services. In: *ICSE*, pp. 604–613. IEEE Computer Society, Los Alamitos (2004)

14. Costa, P., Coulson, G., Mascolo, C., Mottola, L., Picco, G.P., Zachariadis, S.: A Reconfigurable Component-based Middleware for Networked Embedded Systems. *International Journal of Wireless Information Networks* 14(2) (2007)
15. Mottola, L., Picco, G., Sheikh, A.: FiGaRo: Fine-Grained Software Reconfiguration for Wireless Sensor Networks. In: Verdone, R. (ed.) *EWSN 2008*. LNCS, vol. 4913. Springer, Heidelberg (2008)
16. Balasubramaniam, D., Dearle, A., Morrison, R.: A Composition-based Approach to the Construction and Dynamic Reconfiguration of Wireless Sensor Network Applications. In: *Proc. of the 7th Symposium on Software Composition (SC) (2008)*
17. Mozer, M.: Lessons from an Adaptive Home. In: *Smart Environments: Technology, Protocols, and Applications*, pp. 273–298. Wiley, Chichester (2004)

Tackling Automotive Challenges with an Integrated RE & Design Artifact Model

Birgit Penzenstadler

Technische Universität München, Software & Systems Engineering
Boltzmannstr. 3, 85748 Garching, Germany
penzenst@in.tum.de

Abstract. The automotive industry faces the need for large and complex embedded systems. The original equipment manufacturers assign the development of subsystems to suppliers. Therefore they are confronted with many challenges concerning specification, documentation, and integration until start of production.

A major part of these challenges can be tackled with continuous and integrated model-based requirements engineering and design. Following the analysis of the challenges, this paper presents current work on a supporting artifact model for embedded systems development and especially focuses on the design part in more detail.¹

Keywords: Architecture, Design, Documentation, Automotive, Embedded Systems.

1 Introduction

A well-known fact is that the complexity of embedded systems is increasing, especially in the automotive domain, as for example the number of ECUs has increased from less than 10 in 1995 to more than 60 today in upper class cars. Strong crosslinking between them makes designing an overall system architecture even more challenging. The need for appropriate architectural specification and documentation is generally accepted [1]. In the automotive domain, this is complicated by the state of practice distributed development within an association of suppliers.

Contribution. This paper discusses common challenges in software system development in the automotive domain and introduces work in progress on an integrated artifact model for requirements engineering (RE) and design that satisfies the special needs of the automotive domain. The focus lies on the design within the different abstraction layers.

Outline. Sec. 2 briefly sketches the state of practice development process for automotive software and describes the arising challenges and related work. Sec. 3

¹ This work was partially funded by the German Federal Ministry of Education and Research (BMBF) in the framework of the REMsES project.

explains the concepts and structure of the artifact model. Sec. 4 details the design part of the model and shows how to satisfy the mentioned specific needs of the automotive domain. Finally, Sec. 5 proposes possible next steps for the still unsolved challenges.

2 Automotive Software Development Process

The general automotive development process is organized according to the V-model [7]. During the conceptual phase (RE & design), the requirements are elicited, and the logical architecture is designed. Then the technical system architecture, where the components are the electronic control units (ECUs), and networking (i.e., the layout of the wiring harness) are defined, and the software components are specified. The components are either developed in house or assigned to suppliers.

During the realization phase (implementation and integration), the components are implemented and tested, then during integration follow the integration tests, system tests and acceptance tests. The strict deadline is start of production. This whole development cycle entails certain challenges:

Challenge of Architecture Specification. The main aspects for the decomposition or modularization of the system during the conceptual phase are cost-optimization, exchangeability, reliability, and supplier structures. [4] state that “modeling of architectural designs (...) lives on whiteboards, in Microsoft PowerPoint slides, or in the developers’ heads”. Therefore, specification and documentation of an overall system’s architecture is an important challenge.

Challenge of Distributed Development. The highly distributed development in the automotive domain implies the need for thorough requirements specification with adequate interfaces, constraints, and context specifications. A number of constraints arises from different aspects of the system environment and all potentially relevant impact factors have to be specified in the tender documents for the suppliers.

Challenge of Integration. The timespan for coordination and integration until start of production is one to two years, as the distributed development leads to late integration. The plan includes 5 levels of integration from basic physics to serial maturity. There is a strict process for error categorizing, tracing, and solving, with many participants and high costs. Especially for the safety-critical systems the aspect of liability is particularly important.

Related Work. The challenges in the automotive domain have been discussed in depth by [5], while this paper only mentions the challenges that are faced by the artifact model. The challenges identified above have been addressed in part by [9], who explicitly integrate experience-based artifacts during requirements engineering, and by [8,11] on the abstraction level of technical architecture. In contrast, our artifact model covers different layers of abstraction.

3 Artifact Model for RE&Design of Embedded Systems

The REMsES project develops a guidebook with a building set of RE and design techniques tailored for the automotive domain. The key to consistency within the artifacts over the whole development process with distributed suppliers and demanding integration is seamless model-based requirements engineering and design.

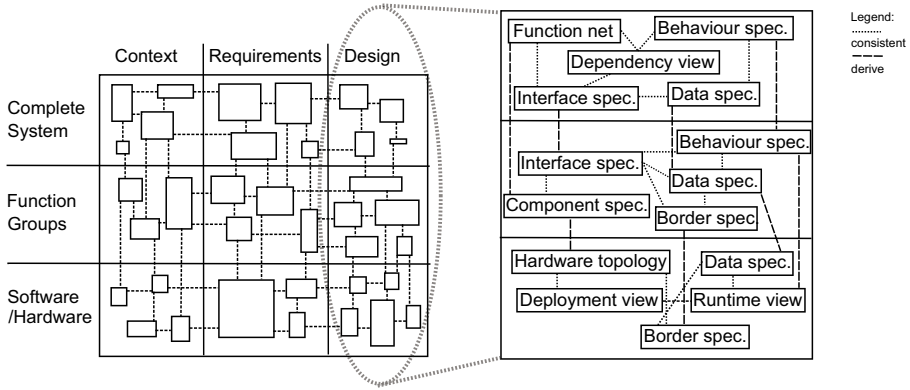


Fig. 1. Dimensions of the Artifact Model and Design Artifacts

The REMsES artifact model [2] is structured in three content categories and three abstraction layers (Fig 1).

Abstraction Layers. The abstraction layers are (top-down) the complete system, the function groups, and the software/hardware (SW/HW), each layer specifying in more detail and adding certain views. On the complete system layer, the system is seen as black box that provides functionality to a user. The function groups layer represents the logical architecture of software subsystems. The SW/HW layer details the technical architecture with little abstraction from the implementation.

Content Categories. The content categories are the context, requirements, and design. The context structures the interrelated conditions of the environment, inter alia with a system vision, business goals, stakeholders, and different types of constraints [12]. The requirements category encompasses documentation and refinement of system goals, use cases, scenarios, and functional requirements [10,3].

The design category captures the early blueprints of the architecture on each layer. Due to space limitations we present only this last category in detail in Sec. 4, as for this workshop the design is the most interesting part of the model.

4 Integrated Design with the Artifact Model

The artifacts for the content category “design” were developed on the basis of [6]. These artifacts (see right side of Fig. 1) are views that are based on one underlying system model per abstraction layer, and they serve as basis for deriving artifacts on lower layers by systematic refinement.

On the *complete system layer*, the design is captured in terms of user services or functions, and the views on those services are a function net, a dependency view, a behaviour view, and interface and data specifications. The function net gives an overview of the structure and interaction of the user functions. The dependency view provides a formal analysis of function interactions and system states, while the system behaviour is modelled with state automata. The data dictionary details on the representation and semantics of the input- and output-data of the functions and the interface captures the communication boundary.

On the *function group layer* the dominant design concepts are logical components and their relations. Different modelling aspects are represented by structural component, interface, data and behaviour views, and an optional border specification. The structure is represented in terms of components that are connected through channels and ports. The interface, data and behaviour views refine the information of the complete system layer per component.

The border specification is a special new artifact that allows for the extraction of a subsystem – its intended use cases are either distributed development by suppliers or reuse within a new surrounding system specification. It encapsulates a short abstract of the functionality and usage of the subsystem and the particularly relevant information from the complete system context for the specific subsystem.

Finally, on the *software/hardware layer* there are hardware topology, runtime view, deployment view, and optional data and border specifications. The hardware topology describes the hardware units of the platform for the technical realization. The runtime view details on the cooperation of hardware clusters and application clusters (= software units) in terms of tasks, events, and buffers. The deployment view maps the application clusters to hardware units. The data dictionary can be refined in case of relevant hardware characteristics, e.g. sensor specifics. In case of reuse or development assignment on this abstraction layer, the border specification is completed with the additional information about technical constraints.

Facing the Challenges

Architecture specification challenges are met by the strong integration of RE and design in the proposed artifact model. The tailoring of the artifact model to the needs of the embedded systems domain becomes more obvious, the lower the regarded abstraction layer is, as the software/hardware layer contains all the specific details from sensor granularity to CAN message codes.

Distributed development is facilitated by the modular structure of the model. The complete system specification has to be maintained only once as the sub-

system specifications are completely integrated. Then the tender documents can be extracted from the overall specification for the assignment to suppliers with the help of the border specification and a guiding process.

Integration is also supported by having one overall systems model, as a major impact on the integrational efforts originates from the quality and adequateness of the system's architecture. The introduced design model facilitates integration in two ways: the artifacts can be used for simulation (verification) and support the process of the V-Model, as the abstraction layers match its steps.

Evaluation has been performed in case studies with driver assistance systems (dynamic window coloring and radio frequency warning) and in student projects for a car light system and a locking system. Furthermore there is a pilot scheme with an instrument cluster being performed by a partner from industry.

5 Conclusion

This paper presents the REMsES artifact model with special emphasis on the design part that meets challenges to current automotive software systems development, namely architecture specification & documentation, development by suppliers, and integration. There will be support for product lines that faces the problems of configurability, which is work in progress from our project partners.

Acknowledgement. Thanks a lot to Doris Wild for helpful feedback.

References

1. Balarin, et al.: A formal approach to system level design: metamodels and unified design environments. In: MEMOCODE 2005, pp. 155–163 (2005)
2. Bramsiepe, et al.: REMsES D2.2: Grobes Produktmodell inklusive Abstraktionsebenen. Project Deliverable (2007)
3. Bramsiepe, et al.: Ableitung von Systemfunktionen aus Zielen und Szenarien. Softwaretechnik-Trends (2008)
4. Brown, A., McDermid, J.: The art and science of software architecture. In: ECSA Conference Proceedings (2007)
5. Broy, M.: Challenges in automotive software engineering. In: ICSE Conference Proceedings, pp. 33–42. ACM, New York (2006)
6. Broy, et al.: Umfassendes Architekturmodell für das Engineering eingebetteter software-intensiver Systeme. Technical report, Technical University of Munich (2008)
7. Kuhrmann, M., Niebuhr, D., Rausch, A.: Application of the V-Modell XT - Report from A Pilot Project. In: Li, M., Boehm, B., Osterweil, L.J. (eds.) SPW 2005. LNCS, vol. 3840, pp. 463–473. Springer, Heidelberg (2006)
8. Lonn, et al.: FAR EAST: Modeling an automotive software architecture using the EAST ADL. IEE Seminar Digests (2004)

9. Paech, et al.: An Experience-Based Approach for Integrating Architecture and Requirements Engineering. In: Proceedings of STRAW 2003 (2003)
10. Pohl, K., Sikora, E.: COSMOD-RE: Supporting the Co-Design of Requirements and Architectural Artifacts. In: RE Conference Proceedings, pp. 258–261 (2007)
11. Tyree, J., Akerman, A.: Architecture decisions: demystifying architecture. *IEEE Software* 22(2), 19–27 (2005)
12. Weyer, T., Pohl, K.: Eine Referenzstrukturierung zur modellbasierten Kontextanalyse im Requirements Engineering softwareintensiver eingebetteter Systeme. In: Modellierung Proceedings, pp. 181–197 (2008)

A Q-Learning-Based On-Line Planning Approach to Autonomous Architecture Discovery for Self-managed Software*

Dongsun Kim and Sooyong Park

Department of Computer Science and Engineering, Sogang University, Shinsu-Dong,
Mapo-Gu, Seoul, 121-742, Republic of Korea
{darkrsw, sypark}@sogang.ac.kr

Abstract. Two key concepts for architecture-based self-managed software are *flexibility* and *autonomy*. Recent discussion have focused on flexibility in self-management, but the software engineering community has not been paying attention to autonomy as much as flexibility in self-management. In this paper, we focus on achieving the autonomy of software systems by on-line planning in which a software system can decide an appropriate plan in the presence of change, evaluate the result of the plan, and learn the result. Our approach applies Q-learning, which is one of the reinforcement learning techniques, to self-managed systems. The paper presents a case study to illustrate the approach. The result of the case study shows that our approach is effective for self-management.

1 Introduction

As software systems face dynamically changing environments and have to cope with various requirements at run-time, they need the ability to adapt to the environments and new requirements[1]. Increasing demands for more adaptive software introduced the concept of self-managed software[2]. Self-management is the means in which the software system can change its composition dynamically without human intervention[3]. To achieve self-management, the system needs to maintain a flexible structure and decide appropriate actions in the presence of new situations. Therefore, flexibility and autonomy at run-time are the key properties of self-management.

In this paper, we focus on achieving the autonomy of architecture-based software systems because recent research already addresses the flexibility of architecture-based software systems by using reconfigurable architectures[4,5]. Specifically, we apply an on-line planning approach to self-management to deal with more autonomous behavior in self-management because the previous approaches so far to autonomy concentrate on designing an off-line planning process where adaptation plans are designed at construction time[3].

* This paper was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea.

This paper proposes an Q-learning[6]-based on-line planning approach to architecture-based self-managed software. To support on-line planning in architecture-based software, this approach presents several elements. Those are (i) representations which describe the current state of a system and possible actions that the system can take, (ii) fitness functions to evaluate the behavior of a system, (iii) operators to facilitate on-line planning, and (iv) a process to apply the previous three elements to actual run-time execution. We also present a case study to verify the effectiveness of the approach.

The paper is organized as follows. The next section gives a brief overview of planning approaches to achieve autonomy in architecture-based self-managed software. Section 3 presents our approach which consists of representation(Section 3.1), fitness(Section 3.2), operators(Section 3.3), and an on-line evolution process(Section 3.4). Section 4 describes a case study conducted to verify the effectiveness of our approach. Section 5 summarizes the contributions of the paper.

2 Planning Approaches in Architecture-Based Self-management

2.1 Off-Line Planning

In architecture-based self-management, off-line planning means that decisions which define the relationship between situations (the current state that a software system encounters) and plans (i.e. architectural reconfiguration actions, e.g. adding, remove, and replacing a component) are made prior to run-time. In other words, whenever a system encounters a specific situation s from an environment each time, the system selects and executes exactly one identical action a . Solutions so far to self-managed systems focus on off-line planning[3]. For example, plans are made by a maintainer through console or hard-coded by a developer[4], components only can restart or reinstall itself when they encounter abnormal states[7], architectural adaptation is described by mapping invariants and strategies in ADL description[8], or architectural changes are triggered by utility functions[5].

These off-line approaches can be effective if developers can identify well-defined goals, states, actions, and rewards along with test environments that exactly illustrate the characteristics of the actual operating environments before deployment time. However, it is very difficult to identify them due to the nature of planning[9] because, in real software development, developers make plans with ill-defined goals, limited numbers of states, actions and partially observable rewards, and test environments poorly describing real environments. On-line planning, which gives more effective autonomy in self-management, presents an alternative to overcome the limitation of off-line planning.

2.2 On-Line Planning

On-line planning in self-management software represents that a software system can autonomously choose an action with respect to the current situation that

the system encounters. Generally, an on-line planning process has three major steps: selection, evaluation, accumulation[9,10]. In the selection step, the system autonomously chooses an action which is suitable for the current situation. Generally, the action is chosen by the greedy strategy in which the best-so-far action is chosen. However, this strategy may lead to the problem of local optima. Hence, the system must adjust its strategy between *exploitation* and *exploration*.

In the evaluation step, the system must estimate the effectiveness of the action which is taken in the selection step. The key issue in the evaluation step is to define the way to determine the numerical values which represent the reward of the action because the numerical representation enables the accumulation and comparison of the rewards.

In the accumulation step, the system stores the numerical values identified in the evaluation step. The system must adjust the accumulation ratio between already accumulated knowledge and newly incoming experience. If the system uses accumulated knowledge too much in the accumulation step, it may slow down convergence speed to optimal planning. On the other hand, if the system uses new experience too much in the step, it may cause ineffective planning.

With these three steps, self-managed software can take advantage of on-line planning in the presence of dynamically changing environments. The next section describes how on-line planning can be applied to actual systems in detail.

3 Q-Learning-Based Self-management

This section presents an approach to designing self-managed software by applying on-line planning based on Q-learning. Generally, we need to consider three elements to apply metaheuristics such as reinforcement learning, simulated annealing, particle swarm optimization; those elements are ‘representation’, ‘fitness’, and ‘operators’[11]. In reinforcement learning, the representations of states and actions are crucial to shape the nature of the search problem. The fitness function is used to determine which solution is better. The operators enable a system to determine neighbor solutions which can accelerate the learning process. Hence, the proposed approach provides state and action representations, fitness function design, and operators to manipulate solutions. In addition to these three elements, the approach provides an on-line evaluation process which describes the control loop of the system at run-time.

3.1 Representation

The representations of states and actions are crucial for designing self-managed software using on-line planning because they define the problem space (situations) and the solution space (architectures) of the system. Instead of intuitive approaches, our approach provides a goal and scenario-based discovery process for more systematic state and action discovery. Goal[12] and scenario-based approaches[13] are widely used for the elicitation of software requirements. Also, goal and scenario discovery have been studied[14].

Table 1. Reformed goals and scenarios

Goal	Scenario	Condition(stimulus)	Behavior(reaction)
Goal 1 Maximize system availability	Sc. 1-1 when the battery of the system is low , turn off the additional backup storage	Cond. 1-1 the battery of the system is low	Beh. 1-1 turn off the additional backup storage
Goal 2 ...	Sc. 2-1 ...	Cond. 2-1 ...	Beh. 2-1 ...
...
Goal n.m ...	Sc. n.m-1 ...	Cond. n.m-1 ...	Beh. n.m-1 ...

The approach exploits goals and scenarios to discover states and actions. Once goal and scenario structure [14] is organized, they can be mapped to states and actions by reforming them. First, scenarios must be reformed into the pair of ‘condition \rightarrow behavior’ or ‘stimulus \rightarrow reaction’, e.g. ‘when the battery of the system is low(condition or stimulus), turn off the additional backup storage(behavior or reaction)’. This reforming is depicted in Table 1. In this table, the discovered goals are listed in sequence (goals will be used to discover fitness functions as described in Section 3.2). The scenarios of a specific goal are listed by the goal. Each scenario is reformed into two additional columns: Condition(stimulus) and Behavior(reaction).

States are identified from the scenarios. Conditions in the scenarios can be candidates of states. A condition represents a possible state of the system, e.g. ‘the system’s battery is low’ implies ‘low-battery’ or ‘the system’s battery is full’ implies ‘full-battery’. A group of conditions represents a dimension(type) of states, for example, ‘battery-level’ is a dimension of state information and it can have value, either ‘low-battery’ or ‘full-battery’. While this information represents a long-term state of the environment, a situation represents a transient change of the environment, e.g. ‘hit by wall’ in an autonomous mobile robot system. Situations are triggers to begin the adaptation of the system. Situations can also be identified from condition information. The condition, which describes a transient event such as ‘when the system is hit by bullet’, can be transformed into a situation. These two pieces of information(situation, long-term state) compose the state information. An example of elements of state information is depicted in the leftmost two columns of Table 2.

Actions can be identified by extracting behavior or reaction from scenarios. As depicted in Table 1, a set of behavior(reactions) is identified in a pair of conditions(or stimuli). If these pairs between conditions and reactions are fixed before deployment time, it can be considered off-line planning, i.e. static plans. The goal of this approach is on-line planning in self-management, the set of actions should be discovered separately. Similar to state information, actions can be identified by discovering action elements and grouping the elements into a type. For example, first, identify action elements such as ‘stop moving’ or ‘enhance precision’. Then, group the elements which have the same type.

Table 2. An example of state and action information

Situation	State(long-term)		Action	
	Type	Range	Type	Range
hit-by-wall	distance	{near, far}	Movement	{precise maneuver, stop, quick maneuver}
hit-by-user	battery	{low, mid, full}	GUI	{rich, normal, text-only}

Examples of action elements and its type are shown in the rightmost column of Table 2. An action element such as ‘rich’ in GUI or ‘stop’ in Movement, implies architectural changes which include adding, removing, replacing a component, and changing the topology of an architecture. Hence, each action must be mapped with a set of architectural changes. For example, the action ‘(Movement=precise maneuver, GUI=rich)’ can be mapped with a sequence of ‘[add:(visionbased_localizer), connect:(visionbased_localizer)-(pathplanner), replace:(normal_gui) -by-(rich_gui)]’ where (...) indicates a component name.

The discovery process shown in this section identifies states and actions by extracting data elements from goals and scenarios. Because goals and scenarios directly represent the objectives and user experiences and also they show how the system influences the environment, state and action information can reflect what the system should monitor and how the system should reconfigure its architecture in the presence of environmental changes.

3.2 Fitness

The fitness function of the system is crucial because it represents the reward of the action that the system chooses. Our approach exploits the goal and scenario structure discovered in section 3.1. In particular, goals are the source of fitness function discovery.

Generally, goals, especially higher ones including the root goal, are too abstract to define numerical functions. Thus, it is necessary to find an appropriate goal level to define the fitness function. It is difficult to define universal rules for choosing appropriate goals which describe numerical functions of the system, but it is possible to propose a systematic process to identify the functions. The following shows the process to define the fitness function.

1. From the root goal, search goals which can be numerically evaluated by a top-down search strategy.
2. If an appropriate goal is found, define a function that represents the goal and mark all subgoals of the goal(i.e. subtree). Then, stop the search of the subtree.
3. Repeat the search until all leaf nodes are marked.

More than one of the fitness functions can be identified by the discovery process. In this case, it is necessary to integrate the functions into one fitness function. The following equation depicts the integrated fitness function:

$$r_t = f(t) = \sum_i w_i f_i(t) \quad (1)$$

where r_t is the reward at time t , $f(t)$ is the (integrated) fitness function, w_i is the weight of the i -th function, $f_i(t)$ is the i -th function of t , and $\sum_i |w_i| = 1$. Equation (1) assumes that the objective of the system is to maximize the values of all functions $f_i(t)$. To minimize a certain value, multiply -1 to the weight value of the function $f_i(t)$. Every function $f_i(t)$ corresponds to the observed data of the system at run-time. For example, the observed network latency $100ms$ at time t is transformed into 10 by the function $f_i(t)$ and multiplied by -1 because it should be minimized.

3.3 Operators

Different metaheuristics use different operators. For example, genetic algorithms use crossover and mutation. The reason why algorithms use operators is to accelerate searching better solutions. In reinforcement learning, operators are used to reduce the search space, i.e. the number of actions. Operator $A(s)$ specifies an admissible action set of the observed state s . For example, when a mobile robot collides with the wall, actions, which are related to motion control are more admissible than those of arm manipulation. This operator is crucial because it can reduce the training time (i.e. learning time) of the system.

3.4 On-Line Evolution Process

With three elements discussed through Section 3.1~3.3, the system can apply on-line planning based on Q-learning. Q-learning [6] is one of temporal-difference learning techniques which is a combination of Monte Carlo ideas and dynamic programming ideas [15]. The reason why we choose Q-learning is its modelless characteristic. This characteristic satisfies the properties of on-line planning described in section 2.2. This section presents the way to exploit those elements in the on-line evolution process. The process consists of five phases: detection, planning, execution, evaluation, and learning phases. The system can dynamically adapt its architecture by executing these phases repeatedly.

Detection Phase. In the detection phase, the system monitors the current state of the environment where the system operates. When detecting states, the system uses the representation of states presented in section 3.1. Continual detection may cause performance degradation. Thus, it is crucial to monitor the change which actually triggers the adaptation of the system. Situations can be appropriate triggers because they describe moments that the system needs adaptation. If a situation is detected, then the system observes long-term states and the current architecture of the system, and denotes them into the representation presented in 3.1. These data are passed to the next phase: the planning phase.

Planning Phase. Using the state identified by the detection phase, the system chooses an action to adapt itself to the state. This phase is related to the selection

step described in Section 2.2 because this phase tries to select an appropriate architectural change with respect to the current situation of the environment. At this phase, the system uses an action selection strategy. In general, Q-learning uses ‘ ε -greedy’ selection strategy as an off-policy strategy to choose an action from the admissible action set $A(s)$ of the current state s . The strategy is a stochastic process in which the system exploits prior knowledge or explores a new action. This is controlled by a value ε determined by the developer, where $0 \leq \varepsilon < 1$. When planning an action, the system generates a random number r . If $r < \varepsilon$, the system chooses an action randomly from the admissible action set. Otherwise ($r > \varepsilon$), it chooses the best-so-far action by comparing the value of each action accumulated in the learning phase. In this manner, the stochastic strategy prevents the system from falling local optima.

Execution Phase. This phase applies the action, which is chosen in the previous phase (planning phase), to the system. As the action describes architectural changes such as adding, removing, replacing components, and reconfiguring architectural topology, the system must have architecture manipulation facilities. Once reconfiguration is done, the system carries out its own functionalities by using its reconfigured architecture. The system keeps executing until it encounters a new situation or it terminates.

Evaluation Phase. This phase is related to the evaluation step explained in Section 2.2 because this phase determines numerical values which evaluate the previous actions (architectural changes) which can be used by the accumulation step (in our approach, the learning phase). After the execution phase, the system must evaluate its previous execution by observing a reward from the environment. As mentioned in section 3.2, the system continuously observes values previously defined by the fitness function. These values will be used for calculating the reward of the action taken.

Learning Phase. In this phase, the system accumulates (as explained in the accumulation step in Section 2.2) the experiences obtained from the previous execution, by using the reward observed in the evaluation phase. This phase directly uses Q-learning. The key activity of Q-learning is updating Q-values. This update process is depicted in equation (2),

$$Q(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a)] \quad (2)$$

where $0 \leq \alpha \leq 1$ and $0 < \gamma \leq 1$. α is a constant step-size parameter and γ is a discount factor. In this manner, the system can accumulate its experience by updating $Q(s_t, a_t) = v$ where s_t is the detected state, a_t is the action taken by the system at s_t , and v is the value of the action on s_t . This knowledge will be used in the planning phase to choose the best-so-far action.

4 Experiment

This section reports on a case study which applies our approach to an autonomous system. The environment in this case study is Robocode [16] which

is a robot battle simulator. Robocode provides a rectangular battle field where user-programmed robots can fight each other.

The reason why we chose Robocode is that it can provide a dynamically changing environment and enough uncertainty, as well as being good for testing self-managed software with on-line planning. In particular, it is hard to anticipate the behavior of an enemy robot prior to run-time. Also, several communities provide diverse strategies for firing, targeting, and maneuvering. These offer opportunities to try several reconfiguration with respect to various situations.

Appendix A provides a robot design which applies our approach. The design includes representations, fitness functions, and operators for the robot.

4.1 Evaluation

This section shows the effectiveness of the approach by presenting the result of robot battles in Robocode.

An experiment was conducted to verify the effectiveness of on-line planning in architecture-based self-management. In this experiment, we implemented a robot based on our approach as described in Appendix A (the robot is denoted ‘A Robot’). Then, we trained the robot for a specific opponent robot (i.e. the opponent robot is an environment of our robot ‘A Robot’). We chose a robot named ‘AntiGravity 1.0’ as the opponent. The opponent (‘AntiGravity 1.0’) is

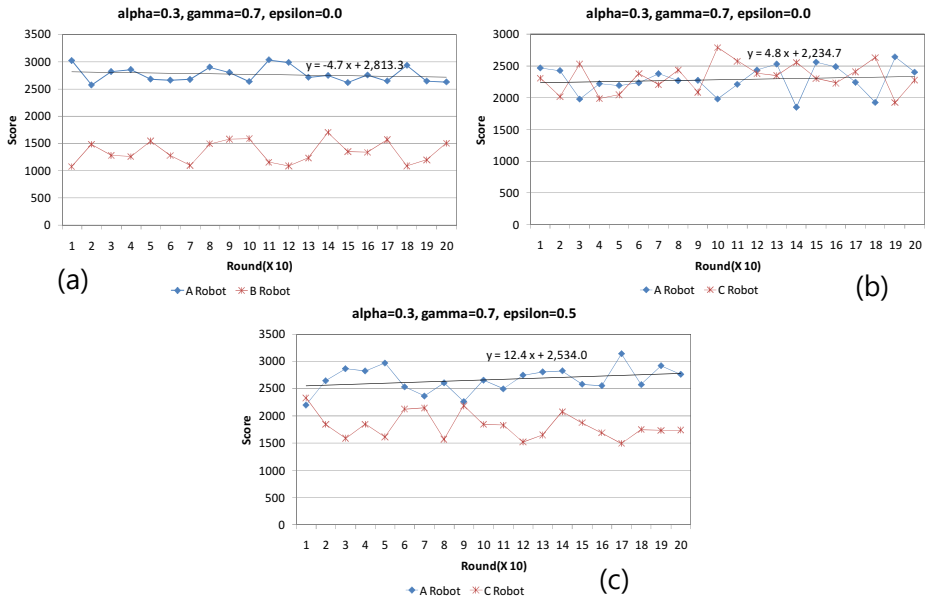


Fig. 1. (a) Exploitation of the previously explored Q-values without further learning. (b) Exploitation of the Q-values to a new robot. (c) Exploration of ‘A Robot’ in the presence of the new robot. X-axis represents rounds that two robots have fought. Y-axis represents scores that each robot obtained.

known for its good performance in battles with many other robots, in Robocode communities. ‘AntiGravity 1.0’ will be denoted by ‘B Robot’. Training is to apply the on-line evolution process described in Section 3.4 to ‘A Robot’ in the battle of two robots (‘A Robot’ and ‘B Robot’). This process made ‘A Robot’ learn the behavior of ‘B Robot’ which is an environment of ‘A Robot’. After training, we investigated the performance of ‘A Robot’ as shown in Figure 11 (a)1. The result shows ‘A Robot’ outperforms ‘B Robot’ and this fact means the approach can effectively make the robot learn the environment.

However, when we introduced a new robot(‘C Robot’ which has ‘Dodge’ strategy), which means the environment is dynamically changed, the robot cannot outperform the enemy as depicted in Figure 11(b). This shows the limitation of off-line planning. To enable on-line planning, we changed ϵ (epsilon) to 0.5 (which means the robot can try new action with respect to the current situation with 50% probability) and the result is shown in Figure 11(c). The result indicates ‘A Robot’ can gradually learn the behavior of ‘C Robot’ and finally outperform the enemy. In other words, it can adapt to the new environment and change its plans dynamically without **human intervention**. This indicates software systems which apply our approach can autonomously search better solutions when they encounters new situations from the environments. The experiment described in this section represents the effectiveness of the approach described in Section 3 by showing that the robot implemented by the approach can learn the dynamically changing environment and outperform off-line planning.

5 Conclusions

Self-managed systems have the potential to provide foundation for systematic adaptation at run-time. Autonomy in self-management is one of the key properties to realize run-time adaptation. To achieve autonomy, it is necessary to provide a planning process to the system. The paper has discussed two types of planning: off-line and on-line planning. On-line planning must be considered to deal with dynamically changing environments. This paper has described an approach to designing architecture-based self-managed systems with on-line planning based on Q-learning. The approach provides a discovery process of representations, fitness functions, and operators to support on-line planning. The discovered elements are organized by an on-line evolution process. In the process, the system detects a situation, plans courses of action, executes the plan, evaluates the execution, and learns the result of the evaluation. A case study has been conducted to evaluate the approach. The result of the case study shows that on-line planning is effective for architecture-based self-management. In particular, on-line planning outperforms off-line planning in dynamically changing environments.

¹ $\epsilon=0.0$ in Figure 11 indicates $\epsilon = 0.0$ in Equation 2. In other words, the robot does not learn the environment no more and Q-value updating converges to best-so-far actions. α and γ indicate a constant step-size parameter (α) and discount factor (γ) and the role of each variable is described in [6][5].

A Robot Implementation

Due to space limitation, we provides an additional material about robot implementation on this URL: <http://seapp.sogang.ac.kr/robotimpl.pdf>.

References

1. Laddaga, R.: Active software. In: Robertson, P., Shrobe, H.E., Laddaga, R. (eds.) IWSAS 2000. LNCS, vol. 1936, pp. 11–26. Springer, Heidelberg (2001)
2. Garlan, D., Kramer, J., Wolf, A. (eds.): WOSS 2002: Proceedings of the first workshop on Self-healing systems. ACM Press, New York (2002)
3. Kramer, J., Magee, J.: Self-managed systems: an architectural challenge. In: FOSE 2007: 2007 Future of Software Engineering, pp. 259–268. IEEE Computer Society Press, Washington (2007)
4. Oreizy, P., Medvidovic, N., Taylor, R.N.: Architecture-based runtime software evolution. In: ICSE 1998: Proceedings of the 20th international conference on Software engineering, pp. 177–186. IEEE Computer Society Press, Washington (1998)
5. Floch, J., Hallsteinsen, S.O., Stav, E., Eliassen, F., Lund, K., Gjørven, E.: Using architecture models for runtime adaptability. *IEEE Software* 23(2), 62–70 (2006)
6. Watkins, C.J.C.H.: Learning from Delayed Rewards. PhD thesis. Cambridge University, Cambridge (1989)
7. Shin, M.E.: Self-healing components in robust software architecture for concurrent and distributed systems. *Sci. Comput. Program.* 57(1), 27–44 (2005)
8. Garlan, D., Cheng, S.W., Huang, A.C., Schmerl, B.R., Steenkiste, P.: Rainbow: Architecture-based self-adaptation with reusable infrastructure. *IEEE Computer* 37(10), 46–54 (2004)
9. Klein, G.: Flexexecution as a paradigm for replanning, part 1. *IEEE Intelligent Systems* 22(5), 79–83 (2007)
10. Klein, G.: Flexexecution, part 2: Understanding and supporting flexible execution. *IEEE Intelligent Systems* 22(6), 108–112 (2007)
11. Harman, M., Jones, B.F.: Search-based software engineering. *Information & Software Technology* 43(14), 833–839 (2001)
12. van Lamsweerde, A.: Goal-oriented requirements engineering: A guided tour. In: 5th IEEE International Symposium on Requirements Engineering (RE 2001), Toronto, Canada, August 27–31, 2001, p. 249 (2001)
13. Sutcliffe, A.G., Maiden, N.A.M., Minocha, S., Manuel, D.: Supporting scenario-based requirements engineering. *IEEE Trans. Software Eng.* 24(12), 1072–1088 (1998)
14. Rolland, C., Souveyet, C., Achour, C.B.: Guiding goal modeling using scenarios. *IEEE Trans. Software Eng.* 24(12), 1055–1071 (1998)
15. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, Cambridge (1998)
16. IBM: Robocode (2004), <http://robocode.alphaworks.ibm.com/home/home.html>

Developing Collaborative Modeling Systems Following a Model-Driven Engineering Approach

Jesús Gallardo, Crescencio Bravo, and Miguel Á. Redondo

Universidad de Castilla-La Mancha, Departamento de Tecnologías y Sistemas de Información,
Escuela Superior de Informática
Paseo de la Universidad 4, 13071 Ciudad Real, Spain
{Jesus.Gallardo,Crescencio.Bravo,Miguel.Redondo}@uclm.es

Abstract. Collaborative modeling systems are useful and promising tools for many tasks. However, they are difficult to build and are domain-specific. In response to this situation, we propose a model-driven process for the development of this kind of systems. This process is based on the use of some ontologies which characterize the concepts used in a software architecture to support collaborative modeling systems. These ontologies, from which the meta-models used in the generation process are derived, are explained in detail. In order to emphasize the utility of the proposal, an example of how the concepts in the ontologies are instantiated in a specific system, SPACE-DESIGN, is shown. It is also explained how by means of this approach it is possible to obtain reconfigurable systems, even at a level of application domain, by means of the use of model specifications and transformations.

1 Introduction and Motivation

Groupware systems are an emergent field of Software Engineering that is gaining importance with time. The new necessities of communication and collaborative work require new types of software [1]. Aspects like awareness, synchronization and shared workspaces are new concepts that groupware introduces. Moreover, groupware is more difficult to design and evaluate than non-collaborative applications because social protocols and group activities must be taken into account [2]. So, groupware development needs specific techniques that consider the aforementioned particularities of this type of software.

In this article, we approach these difficulties by proposing a model-driven engineering approach for the construction of groupware. We focus on a specific type of groupware, which is called general-purpose modeling systems. In these systems, several users typically interact for the construction of a design or artifact, working on a shared space according to the whiteboard metaphor. This design is built according to a specification of a goal or task. With this approach, the use of techniques of software engineering such as meta-modeling is very useful. By means of meta-modeling, we define the structure of the models that the users will build using the system. As we want the structure to vary according to the domain of design, at the greatest level of abstraction we define an ontology that makes up the architectural model of the system. Starting

from this ontology, the meta-models that will be used during the process are derived. These meta-models will be the basis for building the models that characterize a specific tool [3].

The particularity of these systems with respect to other modeling systems is that the design to make is not restricted to a specific domain, that is to say, the system will be able to deal with diverse scopes of design [4] defined by means of a configuration process. This way, our approach avoids the problems of having to re-design the system for each new domain of design. Our meta-modeling approach is going to overcome this and is going to allow us to work with the same system for different domains. Let us think for example about a company of software development that wants their employees to discuss simultaneously on a flow chart and on a class diagram. With our approach, it could be done using the same system, adapting it on each occasion to each specific domain.

This article continues by presenting the foundations of collaborative modeling systems and in particular domain-independent systems. In the third section we explain the concepts identified in the process of meta-modeling. The following section describes an ontological approach to the concepts. Next, we apply our proposal to a case study. Finally, in section 5, we discuss the conclusions drawn and the future work.

2 Domain Independence in Groupware Systems

Within groupware systems, in this work we focus on distributed synchronous systems, and within them, on the particular type of domain-independent modeling systems. The first characteristic of these systems is that their purpose is the development of a model made up by a set of objects and relationships among them. Typically, that model will be built as an answer to a specific goal.

This way, the usual procedure implies that users work on a shared workspace, in which they develop the model in a collaborative way. To achieve this, users participate in design sessions in which they make use of a set of domain objects, predefined objects and relationships that they can locate on this shared workspace. Thus, this situation can refer to a group work activity, if the problem is a real situation to solve in the scope of a company or institution, as well as to an e-learning system, if a learning method based on problem solving is followed.

The second basic characteristic of the kind of systems to be developed is that they are independent of the domain considered. At this point, we can define domain as a certain syntax and semantics that we use to construct our models. Thus, in our systems, the domain over which the model is developed is not fixed, but it can be defined by the user by means of suitable authoring tools.

These systems are going to present some very characteristic problems. Examples of these are the materialization of the shared workspaces, the policies of floor control or turn taking, the processes of coordination and communication, the definition of the domains, etc. In order to show the aforementioned characteristics, some examples of systems related to the approach adopted will now be discussed.

Cool Modes [5] is a cooperative modeling system in which several users work together in order to design a model or artifact. It includes a shared workspace that adopts the metaphor of the electronic whiteboard, in addition to a set of plug-ins that

contain the elements that can be located on the whiteboard. The power of the system is that the constructed model can be simulated, since the plug-ins have this functionality associated. On the other hand, the definition of plug-ins is not very versatile, since they are programmed within the code.

Another example of a similar system is Synergo [6], which is also a design system following a shared whiteboard metaphor. Synergo has several fixed palettes with design components that can be connected to each other. In Synergo the designs are stored in a proprietary format, whereas the different actions are stored in an XML file that can be analyzed later. Another characteristic that Synergo includes is a communication tool (chat) so that the users can discuss among themselves.

In contrast to the described systems, we could also mention some examples of domain-dependent modeling systems, which are systems conceived and designed to work on a certain scope of design. In particular, DomoSim-TPC [7] is one of such systems, since it is aimed to be used on the Domotics domain. Nevertheless, DomoSim-TPC is a modeling system that fulfils the characteristics mentioned, since it has a shared workspace on which the users construct a model as a solution to a previously created problem. Finally, another example is the Co-Lab system [8]. Co-Lab is an environment designed for synchronous collaborative inquiry learning that works with the System Dynamics domain. An important difference between Co-Lab and other modeling systems is that it works with the building metaphor, where each building represents a course in a specific domain. Also, Co-Lab is provided with a number of awareness mechanisms.

As a first conclusion derived from the discussion of these systems, it can be seen that the existing domain-independent collaborative modeling systems do not have as much flexibility as it would be expected. The design palettes are programmed in the code in most of those systems, avoiding end users to extend the functionality by means of the definition of new domains. The solution that we give to this problem is the use of meta-modeling and formal specifications to define the domains, with the use of authoring tools to facilitate this work to end users.

A second conclusion is that domain-specific systems have many more mechanisms of awareness, communication (chat) and coordination than domain-independent ones. The effort made to obtain domain independence seems to be the cause that in other aspects of the system, like awareness, some functionality is lost. Therefore, in the proposal made in this article, domain independence will work together with the use of the mechanisms of awareness, communication and coordination.

3 Ontologies and Meta-modeling for Domain-Independent Modeling Groupware

Within the type of systems we are considering, when working on the organization of models and their definition, it becomes clear that we need to use meta-models to define at different levels the elements on which we will work. In the same way, it will also be interesting to use ontologies to conceptualize our definitions.

This kind of approach leads to the adoption of a model-driven engineering (MDE) approach, which can be the MDA approach that is proposed by the OMG or another kind of model-driven software development process.

At the present time, most methodologies and frameworks that approach the development of groupware are proposing ontologies for the organization of the concepts to be handled, in line with what is being done in the rest of disciplines of Software Engineering. For example, AMENITIES is a methodology based on behavior and tasks models for the analysis, design and development of cooperative systems [9], and it is based on an ontological proposal for the specification of the systems.

Considering the explanations above, our proposal begins with the definition of a series of ontologies that include the concepts that will appear in our domain-independent systems. These ontologies expand the conceptual framework exposed in [3]. In order to do so, we will take advantage of the UML syntax, using class diagrams. These ontologies are going to be used as the basis for the meta-models in our meta-modeling approach, including also the implementation concepts that are not present in the ontologies but are needed in the meta-modeling process.

For doing so, our ontologies are represented as meta-models using a meta-meta-model definition such as MOF or Ecore, depending on the kind of meta-modeling process carried out (MDA or a process that uses the Eclipse meta-modeling plug-ins). Also, model transformations between the different layers of the architecture are made using languages such as ATL (Atlas Transformation Language) or QVT (Query/View/Transformation). This way, models created by instantiating the meta-models obtained stand for the definition of a domain-specific collaborative system made up of a set of collaborative tools. This system is the one generated when reaching the final steps of the process.

Our ontologies are divided into three packages of concepts, which will match three sub-ontologies. First, we will have the domain sub-ontology, which includes the concepts about the domain of design, such as objects, variables and relationships. Within it, a package of a smaller order will contain the graphical aspects that will represent the domain concepts. The next sub-ontology will define the modeling goal in terms of requirements and constraints. Finally, a last sub-ontology is the workspace one, which is divided into two packages: one that deals with the collaborative tasks of the development process and another that deals with the tools that implement those tasks. The ontologies are interrelated among them by means of some concepts that appear in several ontologies. For example, the domain concept is an important one and is present in the domain and modeling goal sub-ontologies.

All the concepts that appear in the ontologies are taken from the analysis of existent groupware tools such as Cool Modes [5], Synergo [6], DomoSim-TPC [7], Co-Lab [8], AMENITIES [9] and some others. Next, the three sub-ontologies will be explained, and their elements, attributes and relationships will be described.

3.1 Domain Sub-ontology

As it has already been mentioned, the domain sub-ontology (Figure 1) contains the concepts related to the domain of work. The domain will indeed be the main concept of this package, with the object and variable concepts also having a special importance. Within this package, another one which contains the graphical aspects is included, that will help us to represent the concepts aforementioned. That package is called domain graphics.

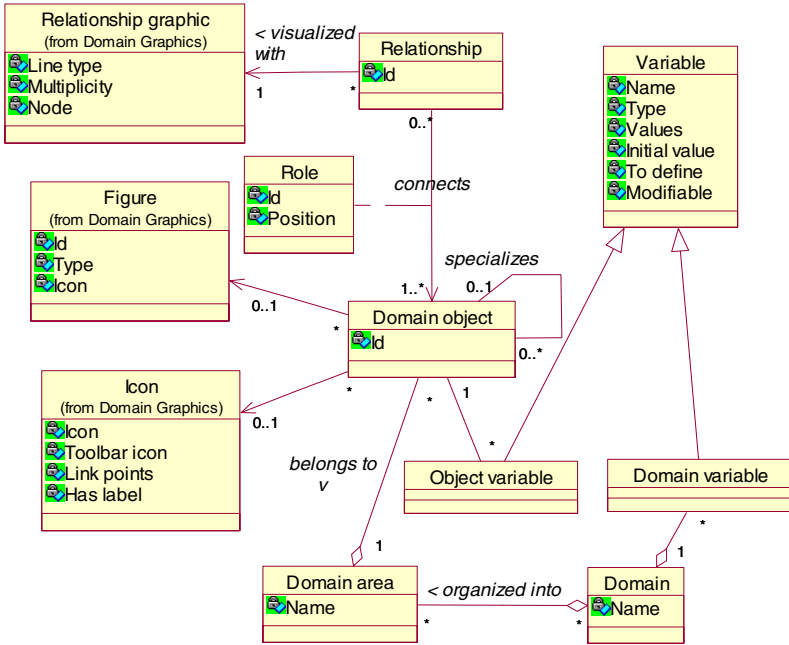


Fig. 1. Domain sub-ontology

The domain sub-ontology is formed by the following concepts:

- **Domain.** This is the main concept of this ontology. A domain of design is made up of the aggregation of a series of areas and a series of global variables.
- **Domain area.** The domain areas are a grouping of objects of the same domain that are related to the same theme.
- **Domain object.** A domain object is an entity that can be instantiated in the model. They are grouped forming domain areas, and will be associated to a specific graphical representation, which can be an icon or a primitive graphic. Also, it is possible for a specific domain object to be a specialization of another kind of domain object.
- **Variable.** A variable is an attribute that can take a certain value. There will be two types: object variables and domain ones. Domain variables are associated directly to the domain, whereas object variables are owned by a certain object within the domain.
- **Relationship.** This is an association between one or more domain objects. The relationship has a multiplicity, which is the number of objects that can participate in it.
- **Role.** In a relationship, every domain object will take a certain role. Thus, we can say that in a relationship there will always be at least two roles, although there can be only one object.
- On the other hand, the domain graphics package is formed by the following concepts:

- Icon. An icon is the graphical representation of an object. It is formed by an image that will be typically read from an image file. An icon also has a representation associated in the toolbar; a set of link points, which will be the places from which a connection could be established to another object; and the possibility of having a text label.
- Figure. This is the graphical representation of an object formed by a primitive figure (circle, rectangle, etc.)
- Relationship graphic. This is the graphical representation of a relationship. It will typically be a line that can have different outlines (continuous, discontinuous, pointed), that could be directed (finished in an arrow or not) and that can have an associated label. If the line is associated to a relationship which has a multiplicity greater than 2, it should have also an associated node. The node will be the place where the lines that come from the participant domain objects come together.

3.2 Modeling Goal Sub-ontology

The modeling goal sub-ontology (Figure 2) addresses the concepts related to the objective to reach in the modeling sessions. Some of these concepts are closely related to the concepts of the domain package, which is the reason why in the diagram the relationships among them have been represented.

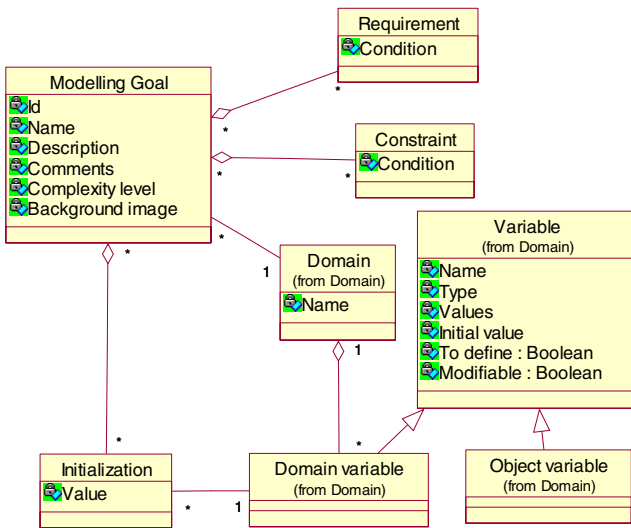


Fig. 2. Modeling goal sub-ontology

The concepts that form the modeling goal sub-ontology itself are the following ones:

- Modeling goal. This is the main concept of the ontology. A modeling goal is made up of a set of initializations, constraints and requirements, and is associated to a specific domain.

- Initialization. This is the initial assignation of a value to a domain variable. The goal can have a series of given initializations.
- Constraint. This is a limitation on some element of the design. For example, on the number of domain objects that can be used to build the model.
- Requirement. This is an obligation imposed on the participants in the design session. Requirements will be directly associated to the modeling goal.

3.3 Workspace Sub-ontology

The last sub-ontology we are going to describe is the workspace one (Figure 3), which describes the tasks that are going to form our collaborative systems and the tools by means of which we will carry out these tasks. Thus, the package is divided into two separated sub-packages; tasks and tools. Outside of those, only the main concept of workspace will be left. These are the concepts that we have described in the workspace sub-ontology:

- Workspace. This is the concept that includes the others. It represents the whole of the system, and it is made up of a set of collaborative tools and associated tasks.
- The next ones are the concepts included in the tasks package:
- Global task. This is each one of the steps in which the development is divided. We will distinguish four tasks, according to the division proposed by Bravo et al. [10]: work distribution, parameterization, design, and simulation.
- Work distribution. This will be the first task of the cycle of work. In it, users will define a list of assignations of actions in order to distribute the work to do.
- Parameterization. In this task, users will give values to the different parameters of the domain. To achieve that, systems based on proposals will be used, and also coordination and collaboration protocols can be useful [11].
- Design. This is the design action itself, in which users will collaborate to design the model according to the restrictions already defined.
- Simulation. This is the final phase of the design process. Users will experiment with the model designed by means of a process of event-based simulation.

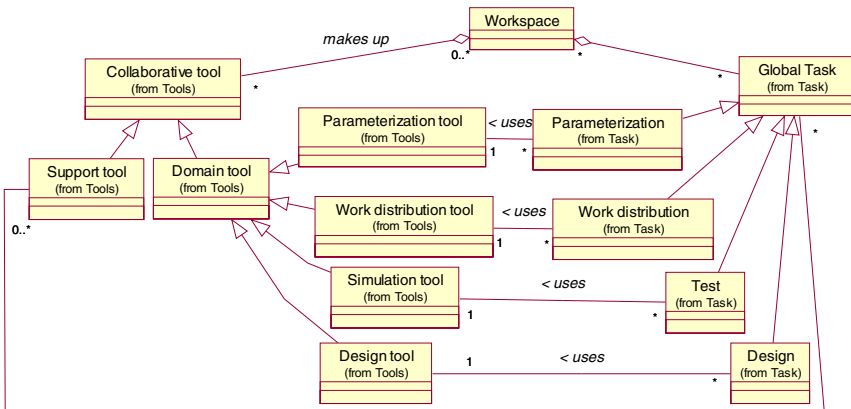


Fig. 3. Workspace sub-ontology

And these are the concepts that form the collaborative tools package:

- Collaborative tool. This represents any type of tool that includes our system. This concept is divided into two categories: support tools and domain tools.
- Support tool. This is a kind of tool that serves as support to the communication and coordination between the members of the design sessions. We have three examples: a chat as communication tool, a session panel that is used to see the connected users, and a coordination tool for turn taking.
- Domain tool. This includes the tools that are used for the implementation of a certain task of the cycle of work in the collaborative system. Therefore, there will be a subtype of domain tool for each type of task.

4 Case Study

As a practical application of our proposal, the SPACE-DESIGN tool has been developed [10]. SPACE-DESIGN is a groupware system that fulfils the characteristics that we have mentioned: it is a tool with support for distributed synchronous work and it allows users to carry out modeling tasks. And, of course, it is domain-independent, since the system reads the domain specification from an XML file and spawns the corresponding user interface in order to make the modeling process.

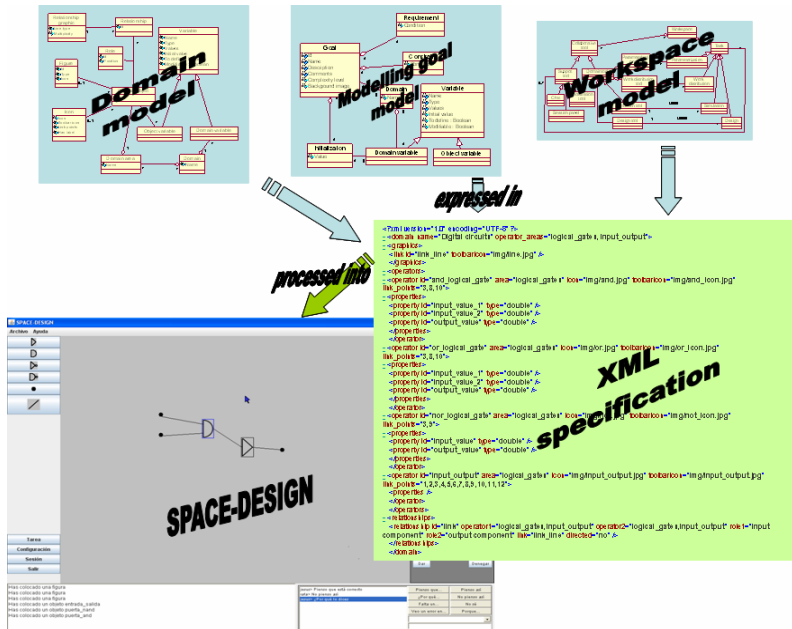


Fig. 4. Correspondence between the models, the specification and the SPACE-DESIGN application

SPACE-DESIGN has been developed following the concepts described in our meta-modeling proposal, and can be applicable to diverse domains of work. However, its development has not been carried out automatically processing the described meta-models, something that must suppose the following evolution of the system.

As has been already said, the SPACE-DESIGN system uses XML for the specification of the domain on which it is going to work (Figure 4). In order to do so, a language for domain specification has been defined. This language describes the different elements that will make up the domain. In the same way, another specification language that describes the goal to reach, with its requirements, constraints and other elements, has been created. These languages are based on the ontologies that we have previously defined, using the same concepts and relating them within the XML file in the same way in which the ontology has defined.

5 Conclusions and Future Work

In this article we have proposed a model-driven engineering approach for the building of domain-independent collaborative modeling systems. This approach can be useful in several fields in which the design of models plays an important role. In order to do that, we have conceptualized the elements of these systems by means of a set of ontologies, which make up the architectural framework upon which groupware tools will be developed. As a practical use of our approach, we have developed the SPACE-DESIGN tool.

The way to specify the application domain in SPACE-DESIGN is similar to the plug-ins in the Cool Modes system. However, it would support one domain at a time, so that by means of an XML-based specification we make the system more flexible and domain-independent. Another advantage of our approach with respect to other tools is the inclusion of explicit mechanisms of awareness and coordination, which have been developed as building blocks.

A next step in our work will be the extension of the conceptual framework of our proposal by defining new sub-ontologies that deal with more concepts present in the groupware tools to be generated. For example, an awareness ontology that includes the different awareness mechanisms in the tools is being developed.

Also, authoring tools that process the specification languages for automatically generating the collaborative systems are being developed. It is currently being studied how these tools, together with the final modeling system, can be integrated into the Eclipse platform. In order to do that, the EMF (Eclipse Modeling Framework) and GMF (Graphical Modeling Framework) plug-ins and the ATL transformation language are being used.

References

1. Guareis de Farias, C.R.: *Architectural Design of Groupware Systems: a Component-Based Approach*. University of Twente, Netherlands (2002)
2. Grudin, J.: *Groupware and Cooperative Work: Problems and Prospects*. In: Baecker, R.M. (ed.) *Readings in Groupware and Computer Supported Cooperative Work*, pp. 97–105. Morgan Kaufmann, San Francisco (1993)

3. Gallardo, J., Bravo, C., Redondo, M.A.: An ontological approach for developing domain-independent groupware. In: Proceedings of the 16th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2007), pp. 206–207. IEEE Computer Society, Los Alamitos (2007)
4. Greenfield, J.: Bare naked languages or what not to model. *The Architecture Journal* 9 (2005)
5. Pinkwart, N., Hoppe, U., Bollen, L., Fuhlrott, E.: Group-Oriented Modelling Tools with Heterogeneous Semantics. In: Cerri, S.A., Gouardères, G., Paraguaçu, F. (eds.) ITS 2002. LNCS, vol. 2363. Springer, Heidelberg (2002)
6. Avouris, N., Margaritis, M., Komis, V.: Modelling interaction during small-groups synchronous problem-solving activities: The Synergo approach. In: Proceedings of the 2nd International Workshop on Designing Computational Models of Collaborative Learning Interaction. Maceio, Brazil (2004)
7. Bravo, C., Redondo, M.A., Ortega, M., Verdejo, M.F.: Collaborative environments for the learning of design: A model and a case study in Domotics. *Computers and Education* 46(2), 152–173 (2006)
8. van Joolingen, W.R., de Jong, T., Lazonder, A.W., Savelsbergh, E.R., Manlove, S.: Co-Lab: research and development of an online learning environment for collaborative scientific discovery learning. In: *Computers in Human Behavior*, vol. 21, pp. 671–688 (2005)
9. Garrido, J.L., Noguera, M., González, M., Hurtado, M.V., Rodríguez, M.L.: Definition and use of Computation Independent Models in an MDA-based groupware development process. *Science of Computer Programming* 66, 25–43 (2007)
10. Bravo, C., Gallardo, J., García-Minguillan, B., Redondo, M.A.: Using specifications to build domain-independent collaborative design environments. In: Luo, Y. (ed.) CDVE 2004. LNCS, vol. 3190, pp. 104–114. Springer, Heidelberg (2004)
11. Gallardo, J., Bravo, C.: Coordination and Communication Protocols for Synchronous Groupware: A Formal Approach. In: Proceedings of the Second IASTED Conference on Human-Computer Interaction. ACTA Press, pp. 55–61 (2007)

Assessing Component's Behavioral Interoperability Concerning Goals

Weimin Ma, Lawrence Chung, and Kendra Cooper

Department of Computer Science
The University of Texas at Dallas
800 West Campbell Road, Richardson, TX 75080, U.S.A.
{weiminma, chung, kcooper}@utdallas.edu

Abstract. As reuse of components becomes increasingly important, so does the assessment of interoperability between them at the time of component assembly. In order for the assembled components to work appropriately according to the needs of the intended stakeholders, it is essential to clearly understand what the individual source components were intended for in the first place. However, research on component interoperability in the past by and large has been focused more on the structural similarities and behavioral interactions between architectural artifacts or between low-level library routines. In this paper, we present an approach to assessing components' behavioral interoperability, with the consideration of the stakeholders' goals which the source components were intended to help achieve. More specifically, we present rules for translating descriptions of stakeholders' goals, together with operations of components and their interactions, into declarative specifications, which are amenable to automatic analysis or automatic generation of visual displays of their execution model. This analysis and visual example will help assess whether the behavior of the assembled components helps, or hurts, the goals of the stakeholders of such assembled components. A Home Appliance Control System is used as a running example to illustrate the approach.

Keywords: Goal, component, behavioral interoperability, declarative description.

1 Introduction

Software components enable practical reuse of software “parts” and amortization of investments over multiple applications. Building new solution by combining bought and made components improves quality and supports rapid development, leading to a shorter time to market [1]. In such a component-based development, selection, evaluation, and integration of components are the key activities that take place early in the life cycle of a component-based system [2]. In order for the assembled components to properly work together, considerable amount of work is required, including the component model's syntactic interface specification, component packaging, and inter-component communication [3]. Meanwhile, the notion of agent and all the related notions (e.g., goals, and plans) are used in all phases of software development, from the early analysis down to the actual implementation [4].

The purpose of component reuse is shortening development time and cost, while achieving stakeholder's goals. For instance, with safety concern in mind when the members of the household are out for vacation, the house owner needs an automatic lighting system to control the household's lighting sequence the same way as members of the household operate the lights to discourage potential intruders. More specifically, the house owner wants a new lighting system to work with an existing central controller which controls the household's appliances. The house owner then asks the manufacturer of the central controller to help choose an automatic lighting system. The manufacturer needs to make sure that not only the central controller works with a new lighting system, but also these two appliances need to work together to achieve the house owner's safety goal. However, research on the past is largely focused on assessing component's structural and behavioral interoperability on low-level library routine and design, without considering whether the integrated components can satisfy the stakeholder's goals or not.

This paper proposes an approach to assessing component's behavioral interoperability to meet stakeholder's goals. The notion of component, i.e., requirements component, in this paper covers interfaces, associated goals and agents as inspired by KAOS [5], NFR Framework [6] and Reference Model for requirements and specifications [7]. The requirements component represents what is observable between the interface and the environment. The behavior of the operations (i.e., interfaces) is represented as state transitions to satisfy goals the components are intended for. Agents and goals are declarative in KAOS [5], i^* [8], Formal Tropos [9], and NFR Framework [6] agent- and goal-oriented approaches. The Alloy language is also declarative [10]. The approach in this paper translates component interfaces, component-intended goals, and the stakeholder's goal into an Alloy model. The translated Alloy model is then automatically analyzed using Alloy satisfiability solver in a visual example. Similar to model checkers, Alloy answers yes if the mode satisfies the properties, and Alloy answers no with a counterexample otherwise. Unlike model checkers, Alloy answers yes with a visual example. When a counter-example is generated for a non-interoperable case, it could well be a case that the counterexample is not really a counterexample, or if it is indeed a counterexample. Similarly, when an example is generated for an interoperable case, it could well be a case that the visual example is not really an interoperable example, or if it is indeed an interoperable example. The user can learn from the visual example/counterexample that either the original model is correct or not, or the properties are correct or not.

This paper benefits from the following related work. Becker and Romanovsky [11] propose adaptation model where adapter is necessary to bridge the gap between two components with provided and required interface using UML [12]. Their interface model is classified into syntax-based interface model, behavioral interface model, interaction-based interface model, quality-based interface model, and conceptual interface model. Schafer and Knapp [13] describe whether the interactions expressed by UML collaboration can be realized by a set of state machines using SPIN model checker. Cubo, Salaun, and et al [14] specify the component as linear transition system, and the context as the communications among the components. The final adapter is derived automatically. Mouakher, Lanoix, and Souquieres [15] propose an automatic translation from UML interface and their associated protocol state machines into an abstract machine notation, i.e., B specification. The automatic verification of

the interoperability is done by the B prover, i.e., mathematical theorem substitution. Supakkul, Oladimeji, and Chung [6] study the non-functional interoperability of components. Component is represented in UML component diagram with provided and required interfaces, and non-functional aspects are represented in Non-Functional Requirements (NFR) softgoals [16]. These works lack for either notion of goal or visual example to confirm the validity of the interoperable components.

The paper is organized as follows. Section 2 provides the background materials on the concepts of goal-oriented approaches, the component, and the model finder Alloy. Section 3 introduces the rules of translating the notion of component into an Alloy model concerning goals for assessing the component interoperability. Section 4 uses the Home Appliance Control System (HACS) example to illustrate the translation with observations. Section 5 gives the conclusions and the future work.

2 Background

As described in the introduction, goals are introduced as part of requirements component. This section briefly introduces KAOS, NFR Framework, and Formal Tropos which foster the notion of goals and the expression of goals in formal notation. A brief introduction of the notion of component and the Alloy model finding technique is also included.

2.1 Goal-Oriented Approaches

KAOS is a framework for goal-oriented requirements acquisition. KAOS focuses on a formal reasoning approach, the objective of which is to enable requirement engineers to automatically derive requirements specifications that satisfy the goals, and to formally verify that goals can be achieved. The appropriateness of a solution is subsequently validated against correctness properties using the proof theory of temporal logic, thus reaching the evaluation state [17]. NFR Framework addresses the issue of representing and analyzing non-functional requirements, and justifying the design decisions with respect to the non-functional requirements. To help address non-functional requirements, the NFR Framework represents softgoals and their interdependencies in softgoal interdependency graphs (SIGs). In NFR Framework, NFR softgoals are satisfied through either AND or OR decomposition of other NFR softgoals or operationalizing softgoals. NFR softgoal as the type is associated with a solution as the topic [6].

Formal Tropos proposes a set of primitive concepts adopted from i^* and a process to build agent-oriented software. The concept of goal can be defined formally in Formal Tropos in terms of a temporal logic inspired by KAOS. In Formal Tropos, goals describe the strategic interests of an actor. Goals have a mode and a set of fulfillment properties. Formal Tropos has two types of dependencies related to a goal, i.e., goal dependencies and softgoal dependencies. Goal dependencies are used to represent delegation of responsibility for fulfilling a goal. Softgoal dependencies are similar to goal dependencies. While it is possible to precisely determine when a goal is fulfilled, the fulfillment of a softgoal cannot be defined exactly [9].

2.2 Component

As indicated in the related work, UML component has been used to represent the notion of component. UML components collaborate through their externally visible interfaces, including required and provided interfaces. Required interfaces represent a dependency that a client component has on certain functionalities that are provided by a server component that implements the functionalities. Provided interfaces represent functional contracts that a component promises to fulfill. Since UML is mainly for functionalities, the notion for non-functional goals is weak.

State transitions with/without formal notations have been used to represent the behavioral aspect of a component. A state represents a snapshot of the behavior of a system whose attributes have certain values. A transition happens when an event occurs and a condition holds for the system's attributes to change values. In other words, a state models a situation during which some invariant condition holds. The invariant may represent a static situation such as an object waiting for certain external event to occur. A state can also model dynamic conditions such as the process of performing some behavior. A transition is a directed relationship between a source state and a target state, representing the complete response of the states to an occurrence of an event of a particular type. State machine diagrams specify state machines, which contain the states and transitions among states [12].

2.3 Alloy

Alloy is a formal language that specifies software systems in a mathematical way by defining its state space, its restrictions and properties. An Alloy model is made of signatures, facts, predicates, restrictions and assertions. A signature declares a set of atoms and a type. Additionally, it is possible to define relations between different signatures in their declaration. Relation can have an arbitrary arity, and can put restrictions on their cardinality. In Alloy, the sets associated with different signatures are disjoint, except when a signature is declared as an extension of another signature.

In Alloy, a "fact" is a formula used to restrict the possible values taken by variables. Additionally, functions and predicates can be defined. Functions are parameterized formulas which return a value. The functions can be used in different contexts by replacing their formal parameters by the actual ones. Predicates are classical logical formulas with parameters. Finally, in an Alloy model, assertions can be defined. Assertions are claims about the model to be validated, or claims implied by the model restrictions. Alloy has two kinds of commands: "check assert" and "run predicate". "Check assert" looks for a counterexample for the assertion, and "run predicate" looks for a model for the predicate. Alloy specifications can be checked by Alloy satisfiability solver. The tool can check the model by running the commands described above. Alloy has small scope hypothesis, which argues that a high portion of bugs can be found by testing the program for all test inputs within some small scope [18].

3 Approach For Interoperability Assessment Concerning Goals

As described in Introduction, the notion of component is requirements component with its specified behavior of the interfaces to achieve the component's associated goals, e.g., the safety goal in this HACS example. Requirements component is defined

as the aggregation of interfaces and goals that the interfaces and their behavior can achieve. The interactions between the requirements components are constrained by the house owner’s safety goal. The house owner’s safety goal, i.e., liveness property in Figure 2, can be expressed using declarative descriptions in temporal logic. The interaction constraints derived from the house owner’s safety goal, i.e., liveness property in Figure 2, is translated together with the specified behavior of the central controller and the lighting system’s interfaces into an Alloy model. The overview of this approach is shown in Figure 1. The translated Alloy model is analyzed using the Alloy satisfiability solver.

3.1 Goal-Oriented Declarative Description Derivation

Goals can be expressed in many ways. For instance, textual descriptions of the house owner’s safety goal can be “when the household are out on vacation, the lights should follow the same sequence as when the members of the household are inside the house”. These imprecise and unstructured descriptions can be expressed in a more organized manner, such as the declarative description using temporal logic. In order to precisely specify a softgoal, e.g., the safety goal for the central controller, the safety goal for the lighting system, and the safety goal for the house owner, a goal-oriented approach such as KAOS or NFR framework can be used to justify the choices of the

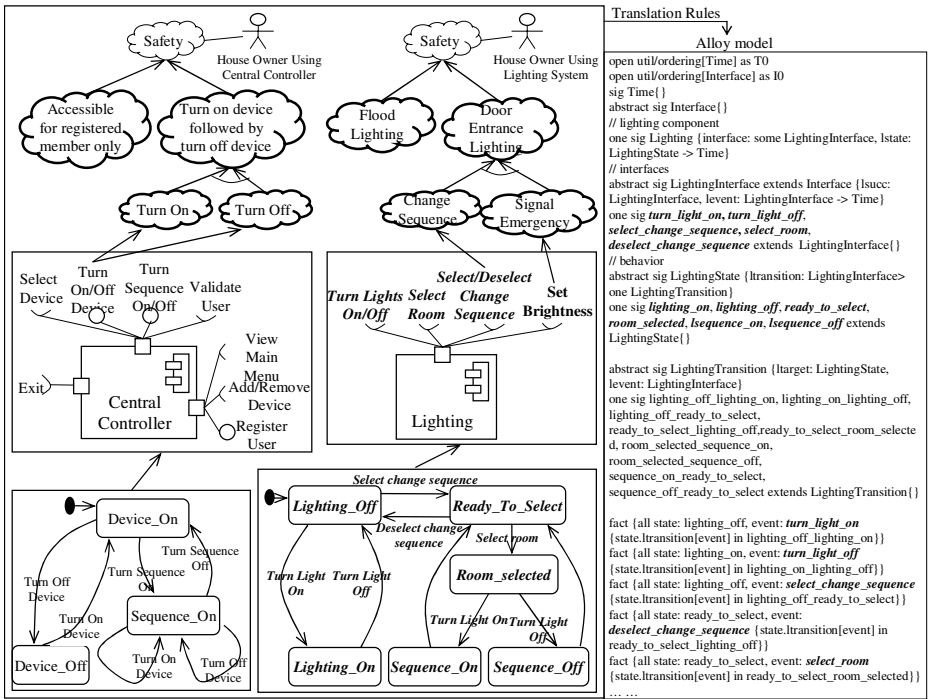


Fig. 1. Overview of translating requirements components into Alloy

declarative description of a goal using temporal logic with respect to components' interfaces, as shown in Figure 2. For instance, the safety goal of the house owner, i.e., liveness property in Figure 2, can be expressed using temporal logic as $G (P \rightarrow X S)$, where G stands for "true all the time", X stands for "true at next time", P is defined as "the members of the household are out", and S is defined as "the household lights change sequence".

A key difference between the approach in this paper and the approach in Formal Tropos is that, this paper considers the analysis of two requirements components together with the stakeholder's goal, while Formal Tropos considers only one similar requirements component. The approach in this paper can be formalized using the following formula, considering central controller and lighting system:

component-interoperability (central controller, lighting) $\Leftarrow \Rightarrow$

{central controller, lighting, interaction(central controller, lighting)} $\models G (P \rightarrow X S)$.

3.2 Interoperability Assessment Using Translation Rules Concerning Goals

In order to take the advantage of the automatic analysis using declarative descriptions and the satisfiability solver that Alloy offers, rules are needed to translate the requirements components and the stakeholder's goals into the declarative descriptions of Alloy. The rules proposed in Table 1 enable the automatic translation. Applying the rules in translating the operationalization of house owner's safety goal (liveness property in Figure 2), interfaces of the central controller and the lighting system and their behaviors, i.e., *turn lights on/off*, *select room*, *select/deselect change sequence*, into an Alloy model is shown on the right side of Figure 1.

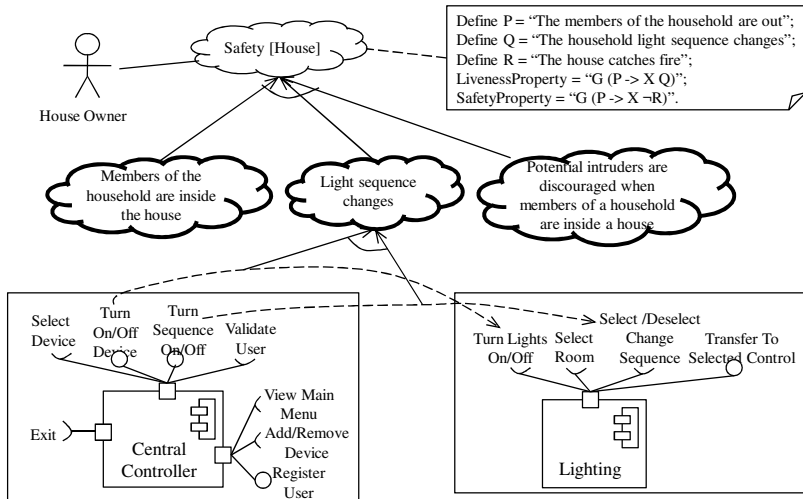
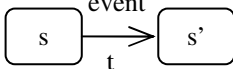


Fig. 2. Example of operationalizing stakeholder's safety goal using component's interfaces and temporal logic

Table 1. Rules for translating requirements components into Alloy

Requirements Component	Alloy
$G (P \rightarrow X Q)$ <i>e.g.,</i> $P = on(turn_sequence)$ $Q = select(change_sequence)$	$pred\ property(i: Interface) \{ t' \ in\ next(t) \mid i.t=P \Rightarrow i.t'=Q \}$ <i>/* "next(t)" is a built-in function in Alloy, which returns a successor of t, or empty set of t is the last element, here t refers to a time point/state. fun nexts (e: elem): lone elem { e.(Ord.next_)}*/</i> <i>e.g.,</i> $pred\ property(i: Interface) \{ t' \ in\ next(t) \mid i.t = turn_sequence \Rightarrow i.t' = change_sequence \}$
$G (P \rightarrow X \neg R)$ <i>e.g.,</i> $P = on(turn_sequence)$ $R = deselect(change_sequence)$	$pred\ property(i: Interface) \{ t' \ in\ next(t) \mid i.t=P \Rightarrow not (i.t'=R) \}$ <i>e.g.,</i> $pred\ property(i: Interface) \{ t' \ in\ next(t) \mid i.t=turn_sequence \Rightarrow not (i.t'=change_sequence) \}$
c: Component <i>e.g., CentralController</i>	$one\ sig\ c\ extends\ Component \{ \}$ <i>e.g., one sig CentralController extends Component { }</i>
i: Interface <i>e.g., TurnOn</i>	$one\ sig\ i\ extends\ Interface \{ \}$ <i>e.g., one sig TurnOn extends Interface { }</i>
	$one\ sig\ t\ extends\ Transition \{ \}$ $one\ sig\ s, s' \ extends\ State \{ \}$ $one\ sig\ event\ extends\ Event \{ \}$ $fact \{ all\ e: event \{ s.transition[e] \ in\ t \} \}, fact \{ all\ trans: Transition \{ trans.target = s' \ and\ trans.event \ in\ e \} \}$
<i>/* The following needs to be declared once */</i> $open\ util/ordering[Time]$ <i>/*"ordering" is a built-in utility in Alloy, which creates a linear ordering over signature Time*/</i> $sig\ Event, Time \{ \}$ $sig\ Transition \{ target: State, event: Event \}$ $sig\ State \{ transition: Event \rightarrow one\ Transition \}$ $sig\ Component \{ state: State, Time \rightarrow one\ State, Time \rightarrow one\ Event \}$ $sig\ Interface \{ \}$	

After operationalizing stakeholder’s goal, i.e., liveness property in Figure 2, into declarative descriptions in temporal logic involving interfaces of requirements components as shown in Section 3.1, the declarative descriptions in temporal logic are translated into an Alloy model. Alloy satisfiability solver is then executed through the command “run predicate” to find whether there is an instance example. An instance example of the execution result can increase the stakeholder’s confidence that the central controller can work with the lighting system to achieve stakeholder’s safety goal. Moreover, a visual example can help the stakeholder to agree/disagree on the execution model. Doing this way, the potential conflicts among the stakeholder’s goal, the interfaces, and the goals of the requirements components, can be revealed. The overview of the approach is shown in Figure 3.

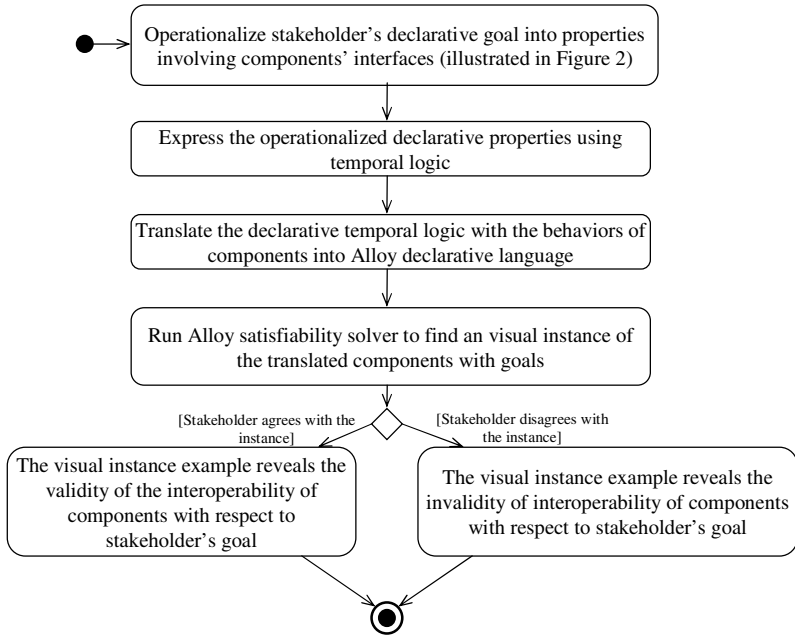


Fig. 3. Overview of behavioral interoperability assessment using the translation rules in Table 1

4 Illustration

Following the scenario in the introduction, the manufacturer of the central controller wants to make sure that its central controller can work with the lighting system to achieve the house owner's safety goal using the approach in Section 3.1. There are three safety goals, the one associated with the house owner and the other two associated with the central controller and the lighting system. By following the approach proposed in Figure 3, the safety goal associated with the central controller and the lighting can be operationalized by their respective operations among their interfaces as shown in Figure 1, such as the operations "turn on/off device" and "select/deselect change sequence", respectively. Moreover, the stakeholder's safety goal, i.e., liveness property in Figure 2, is operationalized using the interfaces of the central controller and the lighting system in temporal logic as: $G(P \rightarrow X S')$, where G stands for "always true". In this example, P represents that the "turn sequence on" button of the central controller is activated, and S' represents that the "select change sequence" button of the lighting system is activated. The stakeholder's safety goal, i.e., liveness property in Figure 2, used as the interaction constraint can be expressed using Alloy declarative language.

4.1 Execution Results

Alloy models are created following the translation rules in Table 1, i.e., from the declarative description of stakeholder's safety goal, i.e., liveness property in Figure 2,

and from the declarative descriptions of central controller and lighting system’s behaviors, i.e., state transitions in Figure 1, into Alloy models. Assumption for this Alloy model is the interfaces of either central controller or lighting system operate one after another in sequence. Besides, skipping an operation of an interface is allowed. For instance, the lighting system operates from “select change sequence” to “turn light on”, and the central controller does nothing. The declarative descriptions of stakeholder’s safety goal, i.e., liveness property in Figure 2, say that the operation “turn sequence on” leads to the operation “select change sequence” or “select room” following a tick of the clock, “turn on device” leads to “turn light on”, and “turn off device” leads to “turn light off”, as illustrated in Figure 2. An instance example as an execution model is generated for this Alloy model as shown in Figure 4.

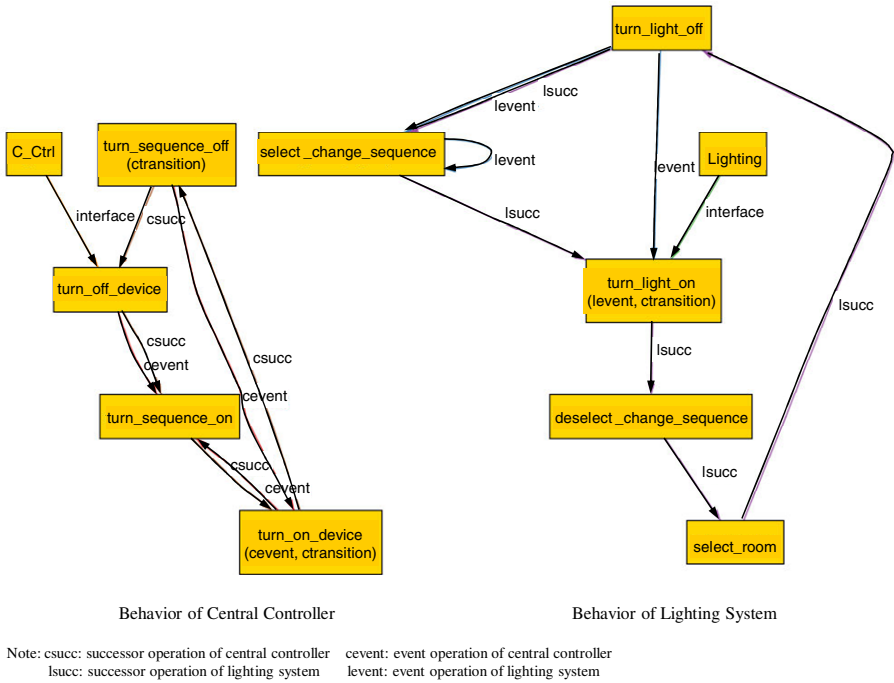


Fig. 4. Example of coordinated behaviors of central controller and lighting system constrained by stakeholder’s safety goal

Depicted in Figure 4, the execution sequence for central controller follows the following transitions, i.e., “turn off device” leads to “turn sequence off”, “turn sequence off” leads to “turn on device”, “turn on device” leads to “turn sequence on”. By referring to the interface behavior of the central controller in Figure 1, this sequence of operations agrees with the house owner’s behavior with certain operations skipped. The execution sequence of the lighting system involves “turn light on” to “select change sequence”, “select change sequence” to “turn light off”, “turn light off” to “select room”, “select room” to “deselect change sequence”, and “deselect change

sequence" to "turn light on". The visual example has the potential to show that the central controller and the lighting system interoperate behaviorally with respect to the stakeholder's goal. After a visual example is generated for an interoperable case, stakeholders can examine it by confirming whether it could be false that the execution model really meets the stakeholder's expectation, or if it is indeed a true execution model that the stakeholder expects.

4.2 Observations

This paper proposes the notion of requirements component and the behavioral interoperability assessment of requirements components using Alloy model finding technique. An HACS example is used for illustration purpose. Translation rules are also proposed from requirements components to Alloy language. The behavioral interoperability assessment by checking the consistency between the goals specified by requirements components and the stakeholder's goal reveals the following observations:

1. There is a difference between components' goals and stakeholder's goal;
2. Declarative description of goals using components' operations/interfaces is amenable to automatic analysis using declarative language and tools;
3. Rules help the translation from requirements components and their interaction constraints into Alloy declarative language;
4. The translated declarative models help assess the consistency between requirements components' behavior with respect to the stakeholder's goal.

5 Conclusions and Future Work

This paper proposes an approach to assess behavioral interoperability of requirements components with respect to stakeholder's goals. This study differentiates the goals associated with requirements components from the goal associated with stakeholder, both of which are specified by behavior of their interfaces. This paper recognizes the similarity between the declarative description of the goals and Alloy declarative language. Therefore, the translation of requirements components and stakeholder's goals into Alloy models is proposed. The stakeholder's goals are expressed using behavior of requirements components' interfaces. The translated Alloy model is analyzed using Alloy satisfiability solver by a visual example to assess the behavioral interoperability between requirements components with respect to the stakeholder's goal. Rules are proposed for the translation. Visual examples can be generated to show that the visual example is truly what the stakeholder expects of the interoperable components, or it is not the case and the stakeholder can discover the inconsistency between components.

The current approach also reveals a number of issues for future research. One concerns translating the requirements components and their interactions into the state transition formalism, then analyzing their interoperability using a model checking technique. Guidelines for interpreting the visual example generated by Alloy are also necessary. Assessing component interoperability concerning agents and goals only is also considered. The consideration of the tool support is underway.

References

1. Szyperski, C., Druntz, D., Murer, S.: *Component Software – Beyond Object-Oriented Programming*. Addison-Wesley Professional/ACM Press (1997)
2. Brown, A.W., Wallnau, K.C.: *Engineering of Component-Based Systems*. In: *Component-Based Software Engineering: Selected Papers from the Software Engineering Institute*, pp. 7–15. IEEE Computer Society Press, Los Alamitos (1996)
3. Rosenblum, D.S., Natarajan, R.: *Supporting Architectural Concerns In Component Interoperability Standards*. IEE Proceedings on Software 147(6), 215–223 (2000)
4. Perini, A., Giorgini, P., Giunchiglia, F., Myloupoulos, J.: *Tropos: An Agent-Oriented Software Development Methodology*. *International Journal of Autonomous Agents and Multi Agent Systems* 8(3), 203–236 (2004)
5. Darimont, R., Delor, E., Massonet, P., Lamsweerde, A.: *GRAIL/KAOS: An Environment for Goal-Driven Requirements Engineering*. In: *Proceedings of the 19th International Conference on Software Engineering*, Boston, Massachusetts, pp. 612–613 (1997)
6. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: *Non-Functional Requirements In Software Engineering*. Kluwer Academic Publishers, Dordrecht (2000)
7. Gunter, C., Gunter, E., Jackson, M., Zave, P.: *A Reference Model for Requirements and Specifications*. *IEEE Software* 17(3), 37–43 (2000)
8. Yu, E.: *Towards Modeling and Reasoning Support for Early Phase Requirements Engineering*. In: *Proceedings of IEEE Symposium of Requirements Engineering*, Annapolis, Maryland, pp. 226–235 (1997)
9. Fuxman, A.D.: *Formal Analysis of Early Requirements Specifications*. Master Thesis, University of Toronto (2001)
10. Jackson, D.: *Software Abstractions: Logic, Language, and Analysis*. The MIT Press, Cambridge (2006)
11. Becker, S., Brogi, A., Gorton, I., Overhage, S., Romanovsky, A., Tivoli, M.: *Towards Engineering Approach to Component Adaptation*. In: Reussner, R., Stafford, J.A., Szyperski, C., et al. (eds.) *Architecting Systems with Trustworthy Components*. LNCS, vol. 3938, pp. 193–215. Springer, Heidelberg (2006)
12. Object Management Group, *OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2*, <http://www.omg.org/docs/formal/07-11-02.pdf>
13. Schafer, T., Knapp, A., Merz, S.: *Model Checking UML State Machines and Collaborations*. *Electronic Notes in Theoretical Computer Science* 47, 1–13 (2001)
14. Cubo, J., Salaun, G., Camara, J., Canal, C., Pimentel, E.: *Context-Based Adaptation of Component Behavioral Interfaces*. In: Murphy, A.L., Vitek, J. (eds.) *COORDINATION 2007*. LNCS, vol. 4467, pp. 305–323. Springer, Heidelberg (2007)
15. Mouakher, I., Lanoix, A., Souquieres, J.: *Component-Adaptation: Specification and Validation*. In: *Proceedings of 11th international Workshop on Component-Oriented Programming*, Nantes, France (2006)
16. Supakkul, S., Oladimeji, E.A., Chung, L.: *Towards Component Non-Functional Integration Analysis: A UML-Based and Goal-Oriented Approach*. In: *Proceedings of 2006 IEEE International Conference on Information Reuse and Integration*, pp. 351–358 (2006)
17. Kavakli, E.: *Goal-Oriented Requirements Engineering: A Unifying Framework*. *Journal of Requirements Engineering* 6(4), 237–251 (2002)
18. Jackson, D., Damon, C.A.: *Elements of Style: Analyzing A Software Design Feature With A Counterexample Detector*. *IEEE Transactions on Software Engineering* 22(7) (July 1996)

A Reference Architecture for Automated Negotiations of Service Agreements in Open and Dynamic Environments^{*}

Manuel Resinas, Pablo Fernández, and Rafael Corchuelo

ETS Ingeniería Informática
Universidad de Sevilla, Spain
<http://www.tdg-seville.info>

Abstract. The provision of services is often regulated by means of agreements that must be negotiated beforehand. Automating such negotiations is appealing insofar it overcomes one of the most often cited shortcomings of human negotiation: slowness. In this article, we report on a reference architecture that helps guide the development of automated negotiation systems; we also delve into the requirements that must automated negotiation systems must address to deal with negotiations of service agreements in open environments. Finally, we analyse how well-suited current software frameworks to develop automated negotiation systems are for negotiating service agreements in open environments. This approach is novel in the sense that, to the best of our knowledge, no previous article compares extensively automated negotiation frameworks in the context of negotiating service agreements in open environments nor provides a reference architecture specifically designed for this scenario.

1 Introduction

Agreements play a major role to regulate both functional and non-functional properties, as well as guarantees regarding the provisioning of a service [12]. Many authors have focused on automating the negotiation of such agreements as a means to improve the efficiency of the process and benefitting from the many opportunities that electronic businesses bring [34].

Negotiating service agreements in an open environment poses a number of specific problems: on the one hand, creating agreements regarding the provisioning of a service involves negotiating on multiple terms, e.g., response time, security features or availability, not only the price, as is the case in typical goods negotiations; on the other hand, open environments require to deal with heterogeneous parties, to negotiate with partial information about them, and to cope

^{*} This work has been partially supported by the European Commission (FEDER), Spanish Government under CICYT project WebFactories (TIN2006-00472), and by the Andalusian local Government under project Isabel (P07-TIC-2533).

with the dynamism inherent to service markets. Examples of open environments include both inter- and intra-organisational systems, e.g., corporate grids.

In this paper, we focus on software frameworks that help develop bargaining systems to negotiate service agreements in open environments.

Building on our analysis of the current literature, we conclude that no current framework is complete regarding the specific problems mentioned above. The goal of this article is to provide a reference architecture that gives a foundation to build such a framework. In Section 2, we detail the specific problems that raise automating the negotiation of service agreements in open environments; in Section 3 we analyse current proposals on automated negotiation frameworks; in Section 4, we outline a reference architecture for automated negotiation of service agreements in open environments to guide the research efforts in this area; in Section 5 we present the conclusions from our analysis.

2 Problems

Automated negotiation systems must cope with the following problems when facing automated negotiations of service agreements in open environments:

1. *Negotiations are multi-term.* Negotiations of service agreements usually involves many terms such as availability, response time, security or price. Therefore, it would be desirable for an automated negotiation system to:
 - (1.1) Support negotiation protocols that allow the negotiation of multiple terms, such as most bargaining protocols.
 - (1.2) Manage expressive agreement preferences: User preferences can be expressed in several ways (usually constraints and rules). Multi-term negotiations require preferences to capture relations between terms and, hence, enable making trade-offs during negotiations.
2. *Parties are heterogenous.* In open environments, parties may implement a great variety of negotiation protocols and present very diverse behaviours during the negotiation. To adapt to this variability, it would be desirable for an automated negotiation system to:
 - (2.1) Support multiple negotiation protocols: Since there is no standard negotiation protocol, different parties may implement different negotiation protocols. Therefore, an automated negotiation system should support several negotiation protocols to avoid losing business opportunities.
 - (2.2) Negotiate the negotiation protocol: Since the best negotiation protocol depends on the concrete situation [5], it is convenient a pre-negotiation phase, in which a negotiation protocol is agreed.
 - (2.3) Support multiple negotiation intelligence algorithms: The effectiveness of a negotiation intelligence algorithm depends on the behaviour of the other parties [5]. Therefore, the automated negotiation system should support several negotiation intelligence algorithms to face the different behaviours of the other parties during the negotiation.

3. *Partial information about parties.* The knowledge about a party is important to strengthen our negotiation capabilities [6]. However, automated negotiation systems usually have only partial information about them [4]. Therefore, it would be desirable for an automated negotiation system to:
 - (3.1) Manage different types of knowledge about the other parties, namely: knowledge about the service or product the other party wish to sell or buy, about the other party itself (*e.g.* its reputation), and about the negotiation behaviour of the party (*e.g.* its temporal constraints).
 - (3.2) Diverse query capabilities: Automated negotiation systems may query information directly to the other party (*e.g.* as a template that should be filled [2]) or they may query third parties to obtain information related to another party (*e.g.* a reputation provider).
 - (3.3) Build analysis-based models of parties: Automated negotiation systems can analyse previous negotiations to build models and, later, use them to make better decisions during the negotiation process [6].

4. *Markets are dynamic.* Service markets can be extremely dynamic because services are not storable, which means that resources not used yesterday are worthless today [7], and, hence, providers may lower the cost of their services when their resources are idle. As a consequence, it would be convenient for an automated negotiation system to:
 - (4.1) Support several negotiations with different parties at the same time, so that the automated negotiation system can choose the party with which the most profitable agreement can be made.
 - (4.2) Select negotiation intelligence algorithms dynamically: Simultaneous negotiations with other parties can have an influence on the negotiation intelligence algorithms employed in a particular negotiation (*e.g.* if a profitable agreement has been found, the system can negotiate more aggressively with the others).
 - (4.3) Support decommitment from previously established agreements: In a dynamic market, new advantageous offers may appear at any time during the negotiations. Hence, it is very convenient to be able to revoke previous agreements, possibly after paying a compensation [8].
 - (4.4) Supervised creation of agreements: To avoid committing to agreements that cannot be satisfied, the automated negotiation system should be supervised by external elements such as a capacity estimator to determine whether an agreement can be accepted or not.
 - (4.5) Build market models: The characteristics of the market may have an influence on the negotiation process [3]. Therefore, it is convenient to build market models to obtain information such as the market reservation price of a service [6].

Last, but not least, a good software framework must be designed with a clear separation of concerns in mind. The separation of concerns indicates how independent is each part of an automated negotiation system from the others. A clear separation of concerns eases the addition of new negotiation protocols or intelligence algorithms without changing the other parts of the system.

3 Analysis of Current Frameworks

Negotiation frameworks focus on the reusability of the different parts of an automated negotiation system. (Note that we focus on software frameworks, not on conceptual frameworks like [9,4].) There are two kinds of negotiation frameworks: protocol-oriented frameworks [10,11,12,13], which deal with the negotiation protocol and interoperability problems amongst automated negotiation systems and intelligence-oriented frameworks [14,7,15,16,17], which focus on the decision-making and world-modelling of automated negotiation systems.

Protocol-oriented frameworks

Kim et al. [11] This article describes a web services-enabled marketplace architecture. It enables the automation of B2B negotiations by defining negotiation protocols in BPEL4WS and developing a central marketplace that runs a BPEL engine that executes the process. The authors also propose a semi-automatic mechanism based on a pattern-based process models to build BPEL negotiation processes.

Rinderle et al. [12] The authors propose a service-oriented architecture to manage negotiation protocols. They define a marketplace, which contains a set of statechart models specifying negotiation protocols. The negotiating parties map the statechart models onto BPEL processes and use them to carry out the negotiation.

Bartolini et al. [10] The authors present a taxonomy of rules to capture a variety of negotiation mechanisms and a simple interaction protocol based on FIPA specifications that is used together with the rules to define negotiation protocols. The authors also define a set of roles that must be implemented to carry out a negotiation process. In addition, the authors define an OWL-based language to express negotiation proposals and agreements.

SilkRoad [13] It consists of a meta-model, the so-called roadmap, intended to capture the characteristics of a negotiation process and an application framework, the so-called skeleton, that provides several modular and configurable negotiation service components.

Intelligence-oriented frameworks

Ashri et al. [15] In this article, two architectures for negotiating agents are described. However, the architecture is described from an abstract point of view and the authors do not provide any details on its components. In addition, it lacks some advanced features to deal with dynamic markets.

PANDA [7] It is a framework that mixes utility functions and rules to carry out the decision-making process. The decision-making component is composed of rules, utility functions and an object pool, which deals with the knowledge about the other parties and the market. However, the object pool is not implemented, but only vaguely specified. Furthermore, it does not support querying other parties to get information; it does not provide mechanisms to change negotiation intelligence algorithms at runtime; it does not

Table 1. Comparison of automated negotiation frameworks (I)

Proposal	(0)	(1.1)	(1.2)	(2.1)	(2.2)	(2.3)
Protocol-oriented frameworks						
Kim et al. [11]	+	+		+		
Rinderle et al. [12]	+	+		+		
Bartolini et al. [10]	+	+		+		
Silkroad [13]	+	+		+		
Intelligence-oriented frameworks						
Ashri et al. [15]	~	+	N/A	+		N/A
Ludwig et al. [16]	~	+	+	+		+
PANDA [7]	~	+	+	+		+
DynamiCS [14]	+	+	N/A	+		+
Benyoucef et al. [17]	+	N/A	N/A	+		+

(0) Clear separation of concerns (2.1) Multiple protocol support
 (1.1) Multi-term negotiation protocols (2.2) Negotiability of protocols
 (1.2) Expressive agreement preferences (2.3) Multiple negotiation intelligence algorithms

support decommitment from previous agreements; and it does not allow a pre-negotiation phase.

Ludwig et al. [16] In this article, a framework for automated negotiation of service-level agreements in service grids is presented. This framework builds on WS-Agreement [2] and provides a protocol service provider and a decision making service provider to deal with the negotiation process. However, this proposal has important shortcomings in dynamic markets since it does not deal with partial information about third parties properly.

DynamiCS [14] It is an actor-based framework, which makes a neat distinction between negotiation protocol and decision making model and uses a plug-in mechanism to support new protocols and strategies. Nevertheless, the framework is not well suited to deal with partial information about third parties and does not cope with dynamic markets.

Benyoucef et al. [17] Their approach is based on the separation of protocols, expressed as UML statecharts, and strategies, expressed as if-then rules. Later, these UML statecharts are transformed into BPEL processes that are executed in a e-negotiation server, and the negotiation strategies are executed by software agents in automated e-negotiation interfaces. In addition, additional services can be composed to complement them. However, how this composition takes place is vaguely defined. Furthermore, the authors do not provide any details on the preferences they manage and whether they support multi-term negotiation protocols or manage different types of knowledge about parties. Another drawback is that it does not seem to be able to build analysis-based models of parties and its capabilities to deal with dynamic markets are limited.

Tables 1 and 2 depict how current negotiation frameworks deal with the problems automated negotiation systems must face in service negotiations in open environments (cf. Section 2): a + sign means that the proposal successfully ad-

Table 2. Comparison of automated negotiation frameworks (II)

Proposal	(3.1)	(3.2)	(3.3)	(4.1)	(4.2)	(4.3)	(4.4)	(4.5)
Protocol-oriented frameworks								
Kim et al. [11]								
Rinderle et al. [12]								
Bartolini et al. [10]								
Silkroad [13]								
Intelligence-oriented frameworks								
Ashri et al. [15]	+		+					
Ludwig et al. [16]		~						
PANDA [7]	+		~	+			+	
DynamiCS [14]								
Benyoucef et al. [17]	N/A	+	~		+		+	

(3.1) Different types of knowledge (4.2) Select dynamically intelligence algorithm

(3.2) Diverse query capabilities (4.3) Decommitment support

(3.3) Analysis-based models (4.4) Capacity factors in binding decisions

(4.1) Simultaneous negotiations (4.5) Market models

dresses the feature; a ~ sign indicates that it addresses it partially; a blank indicates that it does not support the feature; and N/A means the information is not available. The conclusions we extract from this analysis is that protocol-oriented frameworks need to be complemented with decision-making and world-modelling capabilities by means of either ad-hoc mechanisms or an intelligence-oriented framework. Regarding intelligence-oriented frameworks, although current solutions successfully deal with multi-term agreements (1.1 and 1.2) and cope with heterogeneous parties (2.1, 2.2 and 2.3) reasonably well, they lack dealing with partial information about parties (3.1, 3.2 and 3.3) and dynamic markets (4.1, 4.2, 4.3, 4.4 and 4.5).

4 A Reference Architecture for Automated Negotiation Frameworks

To overcome the problems of current negotiations frameworks described in the previous section, we have developed the NegoFAST reference architecture. Its goal is to define the data model, processes and interactions for which an automated negotiation framework should provide support. NegoFAST is divided into a protocol-independent reference architecture, the so-called NegoFAST-Core, and protocol-specific extensions. This allows to deal with a variety of different protocols while keeping the other elements of the reference architecture reusable. In this paper, we just focus on NegoFAST-Core, although a bargaining-specific extension (NegoFAST-Bargaining) has also been developed (more details can be found at <http://www.tdg-seville.info/projects/NegoFAST>).

To describe the NegoFAST-Core reference architecture (*cf.* Figure 1), we decompose it into modules, roles, interactions and environment. Modules are

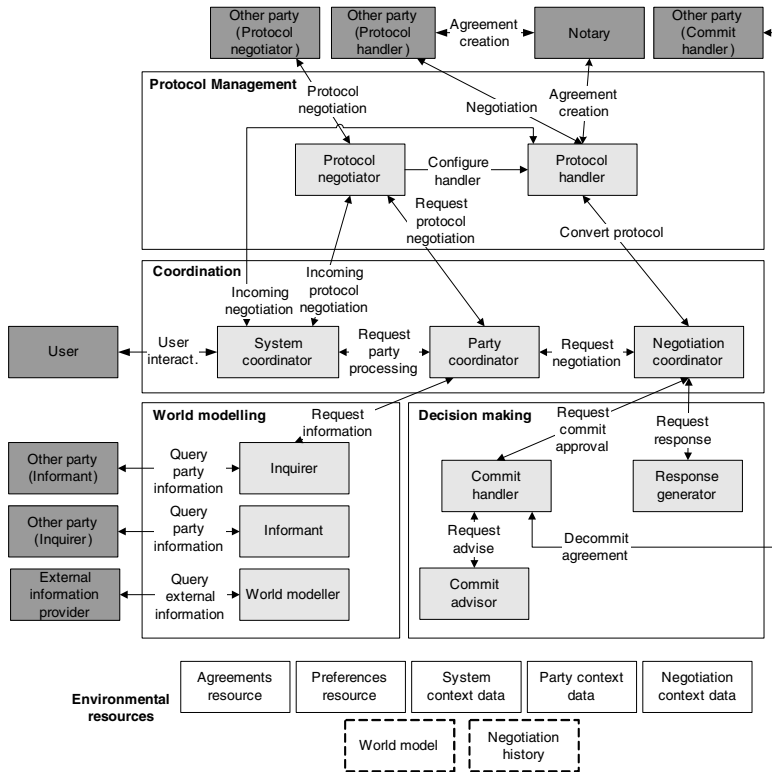


Fig. 1. The NegoFAST-Core reference architecture

depicted as big boxes and are composed of several roles (depicted as small light grey boxes) with arrows connecting them, which represent their interactions. The environment is divided into several resources, which are depicted as white boxes. Finally, elements that are external to the architecture are depicted as small dark grey boxes.

The aim of the protocol management module is to provide the elements that are necessary to deal with the selection and the execution of concrete negotiation protocols and to make the other roles of the architecture independent from them. Protocol management is composed of two roles:

ProtocolNegotiator. Its goal is to select and configure, if necessary, in cooperation with the other negotiating parties, the protocol that will be used during the negotiation process.

ProtocolHandler. It deals with the interaction with the other parties following a negotiation protocol by transforming the syntax of the negotiation protocol into negotiation messages that are understood by the other roles in NegoFAST-Core.

The goal of the decision making module is to provide mechanisms to determine the behaviour of the automated negotiation system during the negotiation. It is composed of three different roles:

ResponseGenerator. Its goal is to determine which messages are sent to the other parties during the negotiation.

CommitHandler. It is responsible for deciding whether and when the system should commit to a proposal and also deciding the decommitment from already established agreements if a more appealing agreement is found.

CommitAdvisor. It analyses the feasibility of accepting an agreement based on domain-specific knowledge (*e.g.* the provider's capacity to provision a proposal) and gives a recommendation.

The goal of the world modelling module is to obtain and manage knowledge about other parties and the market. It is composed of the following roles:

Inquirer. The *Inquirer* is the role in charge of obtaining more information about the other parties by polling their *Informants*.

Informant. It is responsible for publishing all public information that can be useful to other parties in order to evaluate the chances to make an agreement with it.

WorldModeller. Its goal is to build up a model of the other parties together with a model of the market. They are based on information supplied by *ExternalInformationProviders* and previous negotiations.

NegoFAST-Core defines three coordination levels that are coordinated by the three roles of which the coordination module is composed:

SystemCoordinator. It coordinates the interaction with the *User* and the negotiation requests from other parties. Furthermore, it stops the system when a termination condition holds such as reaching a preestablished negotiation deadline or achieving a desired number of agreements. It stores its status in environmental resource *SystemContextData*.

PartyCoordinator. It coordinates the interactions with the other parties before the actual negotiation takes place. This includes getting the information about the party that is necessary to start a negotiation with it by means of the *Inquirer* and agreeing on a negotiation protocol with the other party using the *ProtocolNegotiator*. Its status is stored in environmental resource *PartyContextData*.

NegotiationCoordinator. It coordinates the execution of the negotiation protocol, handled by the *ProtocolHandler* with the decision-making roles of the system. Furthermore, it should be capable of managing several negotiations simultaneously. Its status is stored in environmental resource *NegotiationContextData*.

Additionally, NegoFAST-Core defines the following environmental resources, which are data stores that can be read, modified or both by the roles:

AgreementsResource. It stores all agreements with other parties within the current system context to enable the comparison of agreements already reached with current negotiations and to allow the decommitment of one of them if necessary.

PreferencesResource. It allow the roles in NegoFAST to have access to the user preferences and to evaluate and compare agreements and proposals.

SystemContextData. It stores information managed by the *SystemCoordinator*, which includes: the moment when the system context started, the party references that has been received and the result of their processing.

PartyContextData. It stores information managed by the *PartyCoordinator*, which includes the information gathered by the *Inquirer*; the negotiation protocol selected, and the result of the negotiation.

NegotiationContextData. It stores information managed by the *NegotiationCoordinator*, which includes the current state of the negotiation context and the negotiation messages that have been exchanged with the other parties.

WorldModel. It stores the knowledge the automated negotiation system has about the other parties, the market and the domain the negotiation is about. For instance, knowledge about a the preferences and negotiation style of a party and the market price for a given service.

NegotiationHistory. It stores past negotiations. It is mainly intended for building models based on previous interactions. The *NegotiationHistory* can be seen as a list of the environmental resources of all system contexts that have been processed by the automated negotiation system.

5 Conclusions

Automated negotiation frameworks that help develop bargaining systems to negotiate service agreements in open and dynamic environments. However, current frameworks are not complete with regard to a variety of problems that may arise in such environments (*cf.* Sections 2 and 3). To overcome these issues, we have developed the NegoFAST-Core reference architecture. It provides a founding for the development of negotiation frameworks suited to negotiate service agreements in open environments. NegoFAST-Core can be complemented with protocol-specific extensions such as NegoFAST-Bargaining, which is a bargaining-specific extension we have already developed.

The advantages of having such reference architecture is that it defines the data model, processes and interactions for which an automated negotiation framework should provide support. Furthermore, it provides a common vocabulary to compare different automated negotiation frameworks.

To validate our approach, we have materialised the reference architecture into a software framework. Furthermore, we have used this software framework to implement three different use cases, namely: a computing job submitter, a computing job hosting service and a system to search for equilibrium strategies. The implementations of the software framework and the use cases can be downloaded from <http://www.tdg-seville.info/projects/NegoFAST>.

References

1. Molina-Jimenez, C., Pruyne, J., van Moorsel, A.: The Role of Agreements in IT Management Software. In: de Lemos, R., Gacek, C., Romanovsky, A. (eds.) *Architecting Dependable Systems III*. LNCS, vol. 3549, pp. 36–58. Springer, Heidelberg (2005)
2. Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., Xu, M.: WS-Agreement Specification (2007), <http://www.ogf.org/documents/GFD.107.pdf>
3. Sim, K.M., Wang, S.Y.: Flexible negotiation agent with relaxed decision rules. *Systems, Man and Cybernetics, Part B, IEEE Trans.* 34(3), 1602–1608 (2004)
4. Luo, X., Jennings, N.R., Shadbolt, N., Leung, H.F., Lee, J.H.: A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments. *Artif. Intell.* 148(1-2), 53–102 (2003)
5. Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Wooldridge, M., Sierra, C.: Automated Negotiation: Prospects, Methods and Challenges. *Group Decision and Negotiation* 10, 199–215 (2001)
6. Zeng, D., Sycara, K.: Bayesian Learning in Negotiation. *Int. J. Hum.-Comput. Stud.* 48(1), 125–141 (1998)
7. Gimpel, H., Ludwig, H., Dan, A., Kearney, B.: PANDA: Specifying Policies For Automated Negotiations of Service Contracts. In: Orłowska, M.E., Weerawarana, S., Papazoglou, M.P., Yang, J. (eds.) *ICSOC 2003*. LNCS, vol. 2910, pp. 287–302. Springer, Heidelberg (2003)
8. Sandholm, T., Lesser, V.: Leveled commitment contracts and strategic breach. *Games and Economic Behavior* 35(1), 212–270 (2001)
9. Faratin, P., Sierra, C., Jennings, N.R.: Negotiation Decision Functions For Autonomous Agents. *Int. J. of Robotics and Autonomous Systems* 24(3-4), 159–182 (1998)
10. Bartolini, C., Preist, C., Jennings, N.R.: A Software Framework For Automated Negotiation. In: Choren, R., Garcia, A., Lucena, C., Romanovsky, A. (eds.) *SELMAS 2004*. LNCS, vol. 3390, pp. 213–235. Springer, Heidelberg (2005)
11. Kim, J.B., Segev, A.: A web services-enabled marketplace architecture for negotiation process management. *Decision Support Systems* 40(1), 71–87 (2005)
12. Rinderle, S., Benyoucef, M.: Towards the automation of e-negotiation processes based on web services - a modeling approach. In: Ngu, A.H.H., Kitsuregawa, M., Neuhold, E.J., Chung, J.-Y., Sheng, Q.Z. (eds.) *WISE 2005*. LNCS, vol. 3806, pp. 443–453. Springer, Heidelberg (2005)
13. Strbel, M.: Design of roles and protocols for electronic negotiations. *Electronic Commerce Research* 1(3), 335–353 (2001)
14. Tu, M., Seebode, C., Griffel, F., Lamersdorf, W.: Dynamics: An actor-based framework for negotiating mobile agents. *Electronic Commerce Research* 1(1 - 2), 101–117 (2001)
15. Ashri, R., Rahwan, I., Luck, M.: Architectures for negotiating agents. In: Mařík, V., Müller, J.P., Pěchouček, M. (eds.) *CEEMAS 2003*. LNCS (LNAI), vol. 2691, pp. 136–146. Springer, Heidelberg (2003)
16. Ludwig, A., Braun, P., Kowalczyk, R., Franczyk, B.: A framework for automated negotiation of service level agreements in services grids. In: Bussler, C.J., Haller, A. (eds.) *BPM 2005*. LNCS, vol. 3812, pp. 89–101. Springer, Heidelberg (2006)
17. Benyoucef, M., Verrons, M.H.: Configurable e-negotiation systems for large scale and transparent decision making. *Group Decision and Negotiation* 17(3), 211–224 (2008)

Towards Checking Architectural Rules in Component-Based Design

Sebastian Herold

Clausthal University of Technology - Department of Informatics,
P. O. Box 1253, 38670 Clausthal-Zellerfeld, Germany
sebastian.herold@tu-clausthal.de

Abstract. Current component-based software development (CBSD) approaches rather focus on the design of software systems than on the system's high-level, coarse-grained architecture. They provide modeling techniques to describe the concrete structure of components and their interfaces, how they are connected and how they interact. As an effect of their focus on the design, they are not appropriate to explicitly model the fundamental rules of a software architecture like architectural patterns or reference architectures that restrict the component-based design.

In this paper, we are going to identify some architectural rules in a small example. Furthermore, we will outline how these rules can be used to constrain the component design based upon a modeling approach called DisCComp.

Keywords: Component-Based Software Development, Software Architecture, Architectural Rules.

1 Introduction and Motivation

Component-based software development (CBSD) is broadly accepted as an important paradigm for software development [1]. Systems are no longer developed from scratch, but composed of existing components as well as components to be newly created. Components and their interfaces define the provided and required functionality and form the reusable units of systems in this paradigm. Different CBSD approaches provide modeling and specification techniques for the design and implementation of components and component-based systems. Many of them use UML-like notions or textual specifications to model interfaces and components in a way that abstracts from a certain component implementation technology or a dedicated programming language. Due to their focus on specifying concrete components and interfaces, they usually provide just a small step of semantic abstraction in specifying the system but more a "syntactic" abstraction from implementation.

However, this is inappropriate for the more abstract level of architectures at which a software architect applies architectural rules that form the fundamental structures in a software architecture. At that level, the software architect abstracts from concrete components and their interfaces, and therefore, their concrete functionality. Models and description techniques for CBSD do not capture those architectural rules and how they interact [2].

This particularly holds for our own component modeling approach called DisCComp (a formal model for Distributed and Concurrent Components) [3]. It provides a formal and executable model for component-based systems which enables us to prototype and to simulate components similar to those of common component technologies like EJB or CORBA [4, 5]. In the following, we will give an overview of ongoing and planned work regarding checking the conformance of DisCComp specifications to architectural rules.

2 Foundations of DisCComp

DisCComp is a formal component model based on set-theoretic formalizations of distributed concurrent systems. It allows modelling of dynamically changing structures, a shared global state and asynchronous message communication as well as synchronous and concurrent method calls.

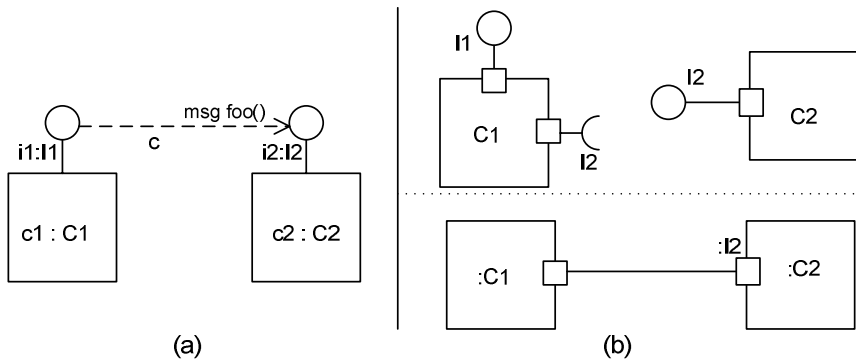


Fig. 1. Illustration of the DisCComp runtime level (a) and specification level (b)

A DisCComp system at runtime consists of instances of different kinds and relations between them. Part (a) of Fig. 1 shows instances of components ($c1$, $c2$), interfaces ($i1$, $i2$), connections (c), and messages (foo). A connection between interfaces is basically a channel for messages or method calls. The formal model defines a corresponding set for each of those different kinds of instances. Instances can be typed, e.g. the type of $c1$ is $C1$.

The relations between instances are reflected by a set of functions, for example, to formalize which interfaces are connected by which connection ($connects(c) = (i1, i2)$) or which component instance an interface is assigned to ($assignment(i2) = c2$).

The formalization enables the runtime environment of DisCComp not just to execute such systems but to check constraints regarding the set of instances. For example, we can postulate the invariant that interface instances of a given type may not be connected with instances of another given type.

Part (b) of Fig. 1 depicts two simplified specifications of single components (upper part). DisCComp specifications are based upon UML component and composite

structure diagrams. The component type C1 requires an interface of type I2. C2 provides I2. A possible system which would result from coupling instances of C1 and C2 would be specified by a composite structure diagram as depicted in the lower part of Fig. 1 (b). By defining multiplicities for ports we are able to define that components provide or require sets of interface instances. Behavioral aspects are modeled in a textual specification language, which is not described here. A more exhaustive description of DisCComp provides [6].

3 Exemplary Architectural Rules

In the following, we will take a closer look to an exemplary architecture, identify two architectural rules, and describe how these rules could be formulated to constrain a DisCComp design model.

3.1 Sample Application

The exemplary application is a typical layered information system as depicted in Fig. 2. There are single layers for presentation, application logic and persisting and managing data. The layer architecture is strict [7] – the presentation may not access the data layer directly.

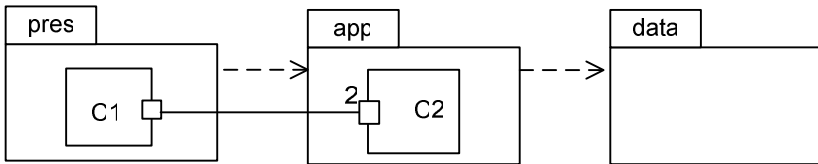


Fig. 2. Layered information system with components

Furthermore, let us assume that the interface of the application layer is service-oriented. On a conceptual level this means that components in this layer provide services which receive and return data via transfer or value objects with copy semantics rather than reference semantics. This means that references from the outside to interfaces inside the application layer are allowed only if they are pointing to interfaces providing services. The interface instance I2 from Fig. 2 is such a “service interface”. Basically in such a system, data from the data layer is processed by a service, copied to transfer objects and handed over to the presentation layer (vice versa for user input). Transfer objects types are also provided by interfaces of application layer components.

These two aspects of the architecture, layered architecture and service-oriented interfaces, influence and constrain the design of single components and their interactions, and therefore constitute architectural rules.

3.2 Architectural Rules in DisCComp

The architectural rules above can be stated as constraints on the design specifications in DisCComp. We can informally express the rule which describes not allowed dependencies between layers as follows:

- Illegal dependency between two layers l_1 and l_2 : For all components, which are owned by an arbitrary layer l_1 , holds that they may not require interfaces provided by components owned by l_2 (see Fig. 2, with “app” for l_1 and “pres” for l_2).

To describe what a service-oriented interface means in DisCComp, we have to clarify what transfer objects are, and what characteristics a method has that realizes a service according to the understanding from above:

- A *transfer object interface* is an interface which can only access its attributes, and which has only associations to other transfer object interfaces. Connected (in terms of the runtime model) interface instances must be assigned to the same component.
- A *service method interface* is an interface which contains only service methods. A service method is a method that uses only transfer object interfaces as parameter and result types. Input transfer objects are copied (see Fig. 3 (b), interface instance “to1_copy”). The copies are assigned to the same component which the called interface is assigned to. Resulting transfer objects are re-assigned to the calling component (see Fig. 3 (c)). The textual DisCComp behavior specification language provides a keyword for assuring this behavior, thus it can be checked during specification.

Thus, we can define the corresponding constraint:

- Layer l_1 has a service-oriented interface: for all components, which are owned by l_1 , holds that all of their interfaces are either transfer object or service method interfaces.

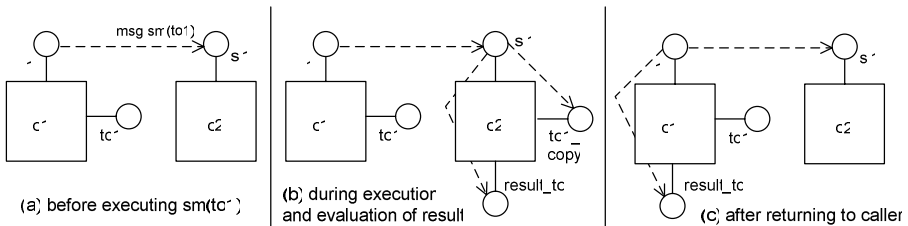


Fig. 3. System states: c_1 calls service method sm_1 at service interface si_1

By checking these constraints during the design of the system, we can assure the consistency between the intended architecture and the developed components, as far as possible. Please notice, that there are architectural rules, or parts of rules, that need checking at runtime. For example, the condition that connected transfer objects are connected to the same component must be checked at runtime. However, we can

check whether a corresponding “runtime constraint” is existing in the specification, which can be interpreted by the runtime environment.

4 Future Work

Most of the work described here are initial ideas that have emerged during the CoCoME project [2], in which we modeled a large component-based system by the means of DisCComp. In that project, a common system was modeled by different teams and different component models. Like most of the teams, the main concern of our work (due to the focus of the approaches) was the design of components and interfaces. The more coarse-grained architecture was given but only followed informally and intuitively.

Currently, we investigate possibilities to formalize architectural rules like those presented in this paper, which are derived from architectural patterns or reference architectures. There exist numerous interesting approaches to design pattern formalization [8]. However, many of those that are closely related to UML extend the UML meta model or provide their own meta models [9, 10] by defining a pattern. We wish to avoid that because of our understanding of an architectural pattern. We regard the solution part of a pattern as a structure of a model (or a part of it) rather than a concept of a modeling language.

First ideas are based on a representation as predicates which are checked against a representation of a design model as a set of facts. Foundations exist in the field of source code or model querying, like for instance [11] or [12]. At the moment, we check architectural rules similar to that described in this paper against Java source code, using a Prolog-like formalization [13].

We are going to realize these checks in the DisCComp specification environment called DesignIt for the proof of concepts. Our goal is to come up with a design tool that supports the designer to create models that are consistent with the intended architectural rules.

References

1. Szyperski, C., Gruntz, D., Murer, S.: Component Software. In: Beyond Object-Oriented Programming, Addison-Wesley, Reading (2002)
2. Rausch, A., Reussner, R., Mirandola, R., Plášil, F. (eds.): The Common Component Modeling Example. LNCS, vol. 5153. Springer, Heidelberg (2008)
3. Rausch, A.: DisCComp: A Formal Model for Distributed Concurrent Components. In: Workshop on Formal Foundations of Embedded Software and Component-Based Software Architectures (2006)
4. Juric, M.B.: Professional J2EE EAI. Wrox Press (2002)
5. Aleksey, M., Korthaus, A., Schader, M.: Implementing Distributed Systems with java and CORBA. Springer, Heidelberg (2005)
6. Appel, A., Herold, S., Klus, H., Rausch, A.: Modelling the CoCoME with DisCComp. In: Rausch, A., Reussner, R., Mirandola, R., Plášil, F. (eds.) The Common Component Modeling Example. LNCS, vol. 5153, pp. 267–299. Springer, Heidelberg (2008)

7. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P.: Pattern-Oriented Software Architecture. A System of Patterns, vol. 1. Wiley & Sons, Chichester (1996)
8. Taibi, T. (ed.): Design Pattern Formalization Techniques. IGI Publishing, London (2007)
9. Maplesden, D., Hosking, J., Grundy, J.: A Visual Language for Design Pattern Modeling and Instantiation. In: Taibi, T. (ed.) Design Pattern Formalization Techniques, IGI Publishing, London (2007)
10. Kim, D.: The Role-Based Metamodeling Language for Specifying Design Patterns. In: Taibi, T. (ed.) Design Pattern Formalization Techniques, IGI Publishing, London (2007)
11. De Volder, K.: JQuery: A Generic Code Browser with a Declarative Configuration Language. In: Van Hentenryck, P. (ed.) PADL 2006. LNCS, vol. 3819, pp. 88–102. Springer, Heidelberg (2005)
12. Störrle, H.: A PROLOG-based Approach to Representing and Querying Software Engineering Models. In: Cox, P.T., Fish, A., Howse, J. (eds.) Proceedings of the VLL 2007 workshop on Visual Languages and Logic, CEUR-WS.org (2007)
13. JQuery – a query-based code browser, <http://jquery.cs.ubc.ca/>

MONET 2008 PC Co-chairs' Message

The research area of mobile and networking technologies applied to the social field has made rapid advances, due to the increasing development of new mobile technologies and the widespread usage of the Internet as a new platform for social interactions. New mobile and networking technologies such as 3G/4G cellular networks and the new generation of wireless local area networks and ad hoc networks are devoted to playing an important role in many areas of social activities, predominantly in those areas where having the right data at the right time is of mission-critical importance. Moreover, mobile and networking technologies for social applications serve groups of people in shared activities, in particular geographically distributed groups who are collaborating on some task in a shared context or independently from their location. As these social applications tend to be large-scale and complex, they have to face difficult social and policy issues such as those related to privacy and security access. By their real nature, mobile technologies can be considered multidisciplinary technologies involving social aspects; indeed they often involve personal, groups and ubiquitous issues, supporting inter-personal connections and involving human-technology interaction in different, and dispersed, contexts. Mobile technologies also play an essential role in personalizing working and interaction contexts, and supporting experimentation and innovation, and the advancement of the field is fundamental for developing social networking. Social networking technologies bring friends, family members, co-workers and other social communities together. These technologies are convergent, emerging from a variety of applications such as search engines and employee evaluation routines, while running on equally diverse platforms from server clusters to wireless phone networks. Networking technologies have to face emerging problems of robustness, such as vulnerabilities to reliability and performance due to malicious attack. The third international workshop on Mobile and Networking Technologies for social applications (MONET 2008) was held in November 2008 in Monterrey. The aim of the workshop is to gather researchers, from academia and industry, and practitioners to discuss new mobile and networking technologies, to identify challenging problems that appear in the social applications of those technologies and to show the results and experiences gathered by researchers. This year, after a rigorous review process that involved 4 referees for each paper, 11 papers were accepted for inclusion in these conference proceedings. The success of the MONET 2008 workshop would not have been possible without the contribution of the OTM 2008 workshops organizers, PC members and authors of papers, all of whom we would like to sincerely thank.

November 2008

Fernando Ferri
Irina Kondratova
Arianna D'Ulizia
Patrizia Grifoni

Personal Sphere Information, Histories and Social Interaction between People on the Internet

Maria Chiara Caschera, Fernando Ferri, and Patrizia Grifoni

CNR-IRPPS,

National Research Council,

Institute of Research On Population and Social Policies, Rome (Italy)

{mc.caschera, fernando.ferri, patrizia.grifoni}@irpps.cnr.it

Abstract. Many people share different pieces of personal information in different on-line spaces, which are part of their social interaction and are points of a trajectory that defines the personal story of each person. This shared information can be used for detecting people and communities with common interests in order to establish their interactions. The great evolution of Web 2.0 and Mobile 2.0 can help individuals to manage and to make available their personal information and social stories from multiple sources and services. In fact, this new communication perspective facilitates the communication among people that share the same interests. Starting from the web information extracted from several on-line spaces it is possible to identify the presence of a specific person in an on-line space. However, associating a specific person to an on-line space for a given event, which identifies one point of the story for that person, can be an ambiguous issue. The paper proposes to use HMMs to model the highest probability for a person to be referred by such specific on-line information.

Keywords: Social interaction, Personal sphere information, Hidden Markov Model.

1 Introduction

Many people share different pieces of personal information in different on-line spaces and this information defines a part of the personal story of each one. However, personal information is distributed and heterogeneous, and it is usually not evident, that information located in different on-line spaces concern the same person. Sometime the same piece of information could be referred to more than one person. In fact, personal information is available in personal websites, blogs, personal pages, e-mail and peer-to-peer file sharing. People communicate using Web 2.0 and Mobile 2.0 services. Each one of the on-line spaces collects different kinds of information in different formats.

On-line personal information could be collected in order to define the story of each person. Considering this information it is possible to analyze the interaction among people for defining sets of people that share interests about work, social activities, cultural activities, sports and hobbies, or that work in the same company.

Collecting web information extracted from several on-line spaces is the starting hypothesis of this paper. In particular, the paper focuses on analyzing how the interaction process can be traced on web space adding information to the personal stories of people with the purpose to correctly manage information about participants at web conferences. The problem of correctly associating information to a person that it refers to correspond to the highest probability for that person to be referred to the considered on-line information. In particular, we propose to add a Hidden Markov Model that permits the unambiguous information-person association in the case of web conferences.

The paper is organized as follows. Section 2 describes the background and related works on methods and models that are used in the literature for assigning a correspondence between the user's behaviors and his/her identification.

Section 3 discusses the scenario in which the different shared personal information can be used and associated to a trajectory identifying a personal story. In section 4 the Hidden Markov Model is presented and defined to build the personal story starting from different events on-line information, with a particular focus on web conferences. Finally, section 5 concludes the paper and discusses the future works.

2 Background

This paper addresses the issue to correctly associate web conferences information to people personal stories. This issue is mainly a classification problem that can be dealt using several methods such as neural networks, principal component analysis, machine learning methods, decision trees, template matching and Hidden Markov Models.

In particular, the interaction process during web conferences can be seen as continuous sequences of group actions that involve simultaneous participants. The interaction actions are not restricted to predefined action sets; therefore, they can be ambiguous and expensive to label.

Therefore, this paper treats this issue using Hidden Markov Models assigning information extracted from on-line conferences to personal stories. In order to present this approach this section explains how these methods have been used in the literature for analyzing interaction sequences.

In detail, several works have focused on supervised learning methods and used statistical models for recognizing communication and interaction sequences. Most existing works are based on HMMs and other probabilistic approaches [1]. In particular, Hidden Markov Models are stochastic models for associating the user input to its correct interpretation. For reaching this purpose in [1] it is specified a sequence of vectors of features extracted from interaction modalities and this sequence is compared with a set of hidden states that can represent the speech word, the letter of the handwritten word or the drawing by the sketch modality according to the definition of the parameters of the model and the modalities to be interpreted. The purpose of this method is to identify the model that has the highest probability to generate the observed output, given the parameters of the model. This approach is well suited for temporally correlated sequential data.

Temporal sequences of human activities are modeled in [2] using a cascade of HMM named Layered Hidden Markov Models (LHMMs). This method analyses states of a user's actions defined by video, audio, and computer (keyboard and mouse)

interactions at various time granularities. In particular, it considers three layers: the first one analyses video, audio, keyboard and mouse activity features; the second level classifies captured features into basic events; and the last one analyses basic events for defining office activities.

Time sequences of actions are also dealt using hierarchical Hidden Markov Models that are multilevel statistical structures [3]. In this case the structural elements of the video are detected using an unsupervised learning scheme for parameters and the model structure. This method consists of two levels of states: the first one represents the semantic events and the second level identifies variations occurring within the same event.

A further approach integrates supervised HMM recognition and unsupervised HMM clustering for analyzing individual and group actions in meetings [4] in a stratified framework.

When it is necessary to deal with temporal data that contain complex structures, such as multiple modalities and interactions among different people, HMMs can be upgraded using Dynamic Bayesian networks (DBNs) [5]. These models use HMMs as input and output and they are widely used to analyze audio and video interactions in social activities. In details, DBNs have been used to pattern meetings as sequence of actions or sentences [6] using multilevel net that are employed to decompose actions as sequences of sub-actions.

However the limitations of the HMMs are connected to: constant statistics within an HMM state; and the independence assumption of state output probabilities [7]. These issues can be overcome capturing the explicit dynamics of parameter trajectories. This purpose is achieved imposing the explicit relationship between static features and dynamic features and translating the HMMs into trajectory models called trajectory Hidden Markov Models [7].

3 Scenario

This work is included in the wide scenario of information sharing. In fact, it starts from information that are acquired from personal websites, blogs, personal pages, e-mail, peer-to-peer file sharing and all information and services available considering Web 2.0 and Mobile 2.0 and that are managed using semantics methodologies. Our purpose is to correctly assign this collected information to personal stories in order to support the detection of people and communities that have common objectives and share common interests in order to allow communication among them (Figure 1).

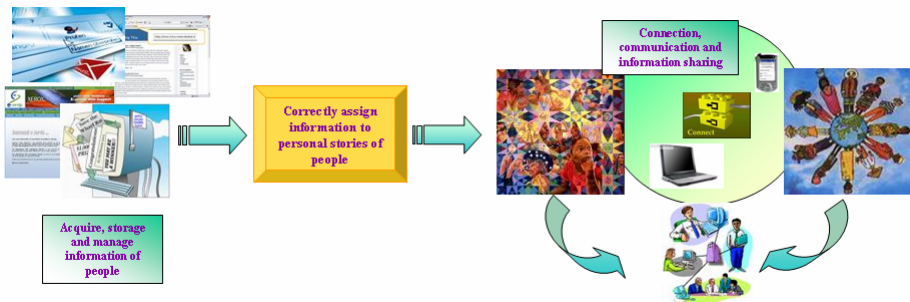


Fig. 1. Scenario

This scenario considers several contexts/places, such as home, work and open space, where users can reach several objectives (to exchange information, to improve social engagement, to support friendship, for entertainment, to develop common interests).

The growing interest on services that enable people to identify trajectories representing personal stories of other people is deeply connected with the development of social networks, the Web 2.0 and Mobile 2.0 services. The availability of information assignable to personal stories can have such interesting implications for simplifying communication in work processes, for security aspects and so on.

Let us suppose we can access to information and data collected by web conferences. This can be obtainable building a “story service”. That is, in order to build the story service, it is necessary to take into account that the same information can be referred to more than one person. At the same time the building process of each story trajectory implies each person has to be identified and associated to his/her trajectory according to a set of information characterizing it.

In particular, below we present an approach for correctly assign information extracted from web conferences to the people personal stories.

In this context information identifying one point of the person’s story trajectory can be ambiguously associated to more than one person. For example we could be interested in providing as service the personal story of a female person whose name is Maria Rossi that has participated at a lot of web conferences in the last year.

That is, we have to consider all the individual actions or group actions and their connected topics, among people participating to web conferences during the last year. The same name in different web conferences could be referred to different people; at the same time different names could be referred to the same person. For example, the story’s trajectory for Maria Rossi can be obtained considering all people whose name is *Maria*, or *Rossi* or *Maria Rossi*, or *Rossi Maria*. However it will probably contain the union of more than one personal story. We can refine the personal trajectory for each person refining information, taking account of the topics of interests arising from the individual and group actions analysis.

In particular, given actions that mainly characterize each person, extracting topics from personal actions in the group (i.e. presentation, discussion), they can be used in order to put an event (a discussion or a presentation) into the story of one person only.

This requires the analysis of the continuous sequences of actions of each person during a web meeting in order to model the structure of each person’s actions during the interaction process. In particular the purpose is to analyze the speech, the gesture and the typing of each person where she/he discusses about specific topics of interests during the web conferences. This analysis aims to define which are the main interests of people that participate in a web conference and which are their modalities of interaction when they discuss about their topics of interests.

Moreover, other information could be used for building each personal story related to the discussed topics, the physical and the virtual location, and so on. A discussion on the involved information is presented in section 4.

Now, from the interaction and communication point of view the person-specific visual features are extracted from the on-line video and the audio features are extracted from the audio of the web conferences.

This information is used to model a set of interaction actions that are commonly found in meetings. In particular these actions include:

- Typing;
- Gesture;
- Speech.

These actions are analysed for each person when she/he discusses on a specific topic and they are grouped in order to model the set of group actions among people during the conferences:

- Presentation: when a person uses speech and/or handwriting and/or gesture using visual images;
- Discussion: when more than one person uses speech and/or handwriting and/or gesture.

These actions are analysed considering different characteristics, specified in the following paragraph, which can represent the participants' behaviours.

4 Building Personal Stories

Building and accessing personal stories of people using web information is an important issue for facilitating contacts and communication among people, between people and organization and among organizations. In fact this information can be used for detecting people and communities that have common objectives and share common interest in order to establish communication among them.

There are many types of information on the Web that can be referred to people and that represent an event in a person life. Videoconferences are one of the most interesting information types available on on-line spaces. The presence of a person in such specific on-line spaces can be described by different descriptive features that allow classifying information. For associating the presence of a specific person in an on-line space, these descriptive features can be modeled by a Hidden Markov Model.

For reaching this purpose it is possible to specify sequences of vectors of features extracted from the on-line spaces potentially referred to a person and to compare this sequence with a set of hidden states that can represent other potential people with the same name.

The purpose of this method is to identify the person that has the highest probability to be referred by the on-line information. In fact, the HMM approach is well suited for temporally correlated sequential data such as information related to events in the life of a person. The participation in a web meeting or a web conference represents an event registered by the Web that can be assigned to the personal history of the people involved in.

In particular, relevant pieces of information that can be used to characterize each person involved in the web conference are:

- The set of individual dialog actions in the meeting or a web conference, such as typing in the chat, speech and gesture;
- Coordinated actions among people such as discussion and presentation (group actions);
- Roles in the coordinated actions (for example presenter, or a listener and so on);
- Interaction patterns;

- Topics of interests, which are dealt in the conference;
- Temporal information, which specifies the time in which the event happened and this information can be explicitly or implicitly registered;
- Spatial information that specifies the location in which the event happened and also spatial information can be explicitly or implicitly registered, moreover it can be referred to a physical or a virtual location;
- Identifiers (such as e-mail address, affiliation etc.) that can significantly contribute to identify the person related to the event.

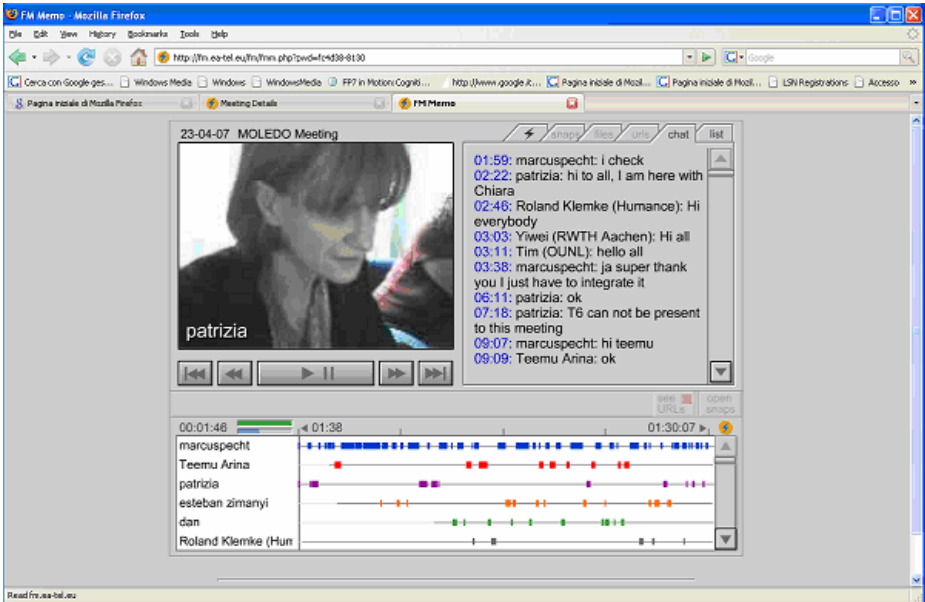


Fig. 2. Web conference/meeting data

Figure 2 shows a typical web application to do web meetings and similar applications that can be used for videoconferences. All these applications contribute to enrich the on-line presence of information concerning people.

In detail, an extractor of information from multimodal data manages audio-video data of the conference, identifying the previously listed kind of characteristics of the individual and collective participation in the web meeting or conference. Each kind of information (i.e. presentation) allows characterizing behavioural patterns of participants in relation to this specific event. In this sense, the participation of a person to the event can be analyzed along the temporal dimension (sequence of dialog actions) and characteristics of each dialog action. These details allow recognizing similar behavioural patterns of people (with the same name) participating in this kind of event in order to associate different events to the same person.

Figure 3 shows the approach that permits to associate events (actions in a web conference) with a person. We have that the web conference data are extracted considering the sequence of dialog actions to analyze the listed set of characteristics and to build behavioural patterns.

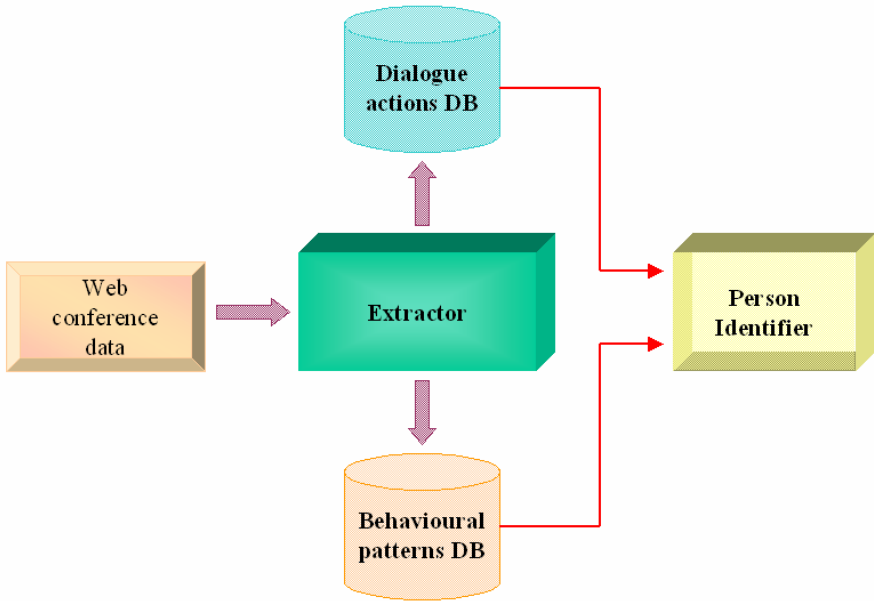


Fig. 3. Description of the approach

All these information contribute to the identification of the specific by means the *Person Identifier* module that analyze different events (and information) on the Web in order to identify sequences of events and information that can be brought back to the same person. This information adds a piece to the personal story of each person.

The *Person Identifier* module is based on an HMM that finds participation and brings together events for each person of the conference. It contains as observation sequences information about participations of people to events described in terms of:

- The set of individual dialog actions;
- Group actions;
- Roles in the group actions;
- Interaction pattern;
- Topics of interests;
- Temporal information;
- Spatial information;
- Identifiers.

Moreover we assume that the hidden states of the model are defined by the name of people.

Therefore the structure of the HMM is:

- Hidden states: $S=\{U_1, U_2, \dots, U_n\}$: people that are possible members of the web conference (described in terms of the several characteristics that can be matched to the information about participation in events);

- Visible states: $Q = \{O_1, O_2, \dots, O_j\}$: information about participations of people in web meetings or conference (i.e. the sequence of dialog actions of a person during the event);
- HMM parameters θ :
 - o Initial probability distribution matrix $\pi = \{P(U_1), P(U_2), \dots, P(U_n)\}$: probability that the person U_1 or U_2 or... U_n is the member of the conference at time 1 respectively;
 - o Matrix of transition probabilities: $A = [a_{ij}]_n \times_n$ that defines all possible transitions among hidden states;
 - o Matrix of emission of probabilities $B = [b_{sk}]_n \times_j$ that defines the probabilities that sequence k of information about participations of people in web meetings or conference is related to the member of the web conference s .

In this manner the Person Identifier can build sequence of events related with a high degree of probability to the same person.

5 Conclusions and Future Work

The availability of a great amount and variety of information on the Internet, and the arising of Web 2.0 and Mobile 2.0 services is producing a great interest on the possibility to share information in order to detect people and communities with common interests and establish their interactions.

Information available to build personal stories is available. However, in order to build the stories service, it is necessary to take into account that the same information can be referred to more than one person. At the same time the building process of each story trajectory imply each person has to be identified and associated to his/her trajectory according to a set of information characterizing it.

The paper has presented the particular issue of access to information and data collected by web conferences and the definition of stories for the web conferences participants assigning information extracted from web conferences to the people personal stories. We have used a Hidden Markov model to define the highest probability for a person to be referred by such specific on-line information. As a future work we are developing a pilot service in order to widely test the assignment correctness of on-line events and information with each person, and consequently the correct personal story definition.

References

1. Murphy, K.: Dynamic Bayesian networks: Representation, inference and learning. Ph.D. dissertation, UC Berkeley (2002)
2. Oliver, N., Horvitz, E., Garg, A.: Layered representations for learning and inferring office activity from multiple sensory channels. In: Proc. ICMI, Pittsburgh (October 2002)
3. Xie, L., Chang, S.-F., Divakaran, A., Sun, H.: Unsupervised discovery of multilevel statistical video structures using hierarchical hidden markov models. In: Proc. ICME (July 2003)

4. Zhang, D., Gatica-Perez, D., Bengio, S., McCowan, I., Lathoud, G.: Multimodal group action clustering in meetings. In: Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks, New York, USA, October 15, 2004, pp. 54–62 (2004)
5. Al-Hames, M., Rigoll, G.: A multi-modal mixed-state dynamic Bayesian network for robust meeting event recognition from disturbed data. In: Proc. IEEE ICME (2005)
6. Dielmann, A., Renals, S.: Dynamic Bayesian networks for meeting structuring. In: Proc. ICASSP (2004)
7. Zen, H., Tokuda, K., Kitamura, T.: Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences. *Computer Speech & Language* 21(1), 153–173 (2007), <http://dx.doi.org/10.1016/j.csl.2006.01.002>

Discovery of Social Groups Using Call Detail Records

Huiqi Zhang and Ram Dantu

Department of Computer Science and Engineering
University of North Texas
Denton, TX 76201 USA
{hz0019, rdantu}@unt.edu

Abstract. In this paper we propose the affinity model for classifying social groups based on mobile phone call detail records. We use affinity to measure the similarity between probability distributions. Since phone calls are stochastic process, it makes more sense to use probability affinity to classify the social groups. This work is useful for enhancing homeland security, detecting unwanted calls (e.g., spam) and product marketing. For validation of our results, we used actual call logs of 100 users collected at MIT by the Reality Mining Project group for a period of 8 months. The experimental results show that our model achieves good performance with high accuracy.

Keywords: Social groups, call detail records, Hellinger distance, affinity.

1 Introduction

Social groups can be defined as sets of people who have common interests, share their experience, express similar ways of thinking and reacting, share the same opinions, do similar things and have the same goals.

There are various applications for social groups. For example, in marketing, if someone buys something, his family members and friends are likely to have the same interests to buy the same or a similar thing and have a similar level of income although we do not know how much they earn. So we may find potential buyers by social groups. Another important application for social groups is in the area of national security. For example, if somebody is a terrorist or robber, his intimate friends or socially close communication partners are likely (not necessary) to be terrorists or robbers too, since no law-abiding person wants to have some friends who are terrorists or robbers. One more application is used to quantify the telecommunication presence. On different days and at different times people usually would like to communicate with different groups of people. For example, we prefer to communicate with our colleagues in work time and to communicate with our family members, relatives and friends in non-work time. Further, in our busy hours we only would like to have necessary communications with our socially close members such as family members, bosses and others. Additionally, we may enhance detecting unwanted calls (e.g., spam) by social groups. For example, the spammers are definitely socially far from us. If we are not sure that the incoming calls which come from socially far

members are spam or not, the system may not let the phone ring and forward the calls to the voice box automatically.

Most social network research and social relationship analysis are based on blogs, emails or the World Wide Web [1- 26]. Since mobile phones have become the main communication media for people in recent years, some researchers' interests in social networks concentrate on social relationship analysis based on call detail records [27-37].

In this paper we propose the affinity [38] model to classify social groups. In Section 2 we briefly review the related work. In Section 3 we describe the model and method for social groupings. We performed the experiments with actual call logs and discuss the results in Section 4. We describe the validation of our model, conducted by the actual call logs, in Section 5. Finally, we have the conclusions in Section 6.

2 Related Work

A social network is defined as a set of actors (individuals) and the ties (relationships) among them [1]. There are two fundamental interests in social networks: the relational ties and the actors. In [2] the author summarized the social network measures involving the relational ties, the actors, and the overall consequences of the network topology. In [3] the authors proposed a concept describing a composite network that incorporates the multi-dimensionality of interpersonal relations is the meta-matrix.

In [4] the author discusses the strength of social relations between two persons measured with the email conversation. By the Loom system in [5] users can visualize social relationships on the basis of Usenet conversation. In [6] the several text analysis procedures are used to compute a net and visualizations of social relations among authors. Microsoft Netscan [7] is used for searching, visualizing and analyzing newsgroups. The various visualization techniques and extensive relation analysis and content analysis are combined to allow for an improved navigation in the Usenet. The Friend-of-a-Friend (FOAF) project [8] explores the application of semantic web technologies to describe personal information: professional and personal lives, their friends, interests and other social relations. In [9, 10] the automatic detection methods of subgroup structure of social networks are investigated based on expected density and edge betweenness. In [11] the authors use block technique, which focuses on the pattern of connectivity to find clusters of relationships that might be hidden. In [12] the generalized block modeling method was proposed to enable partitions of smaller, more precise block types and predefined block structures based on attribute data. In [9, 13, 14, 15, 16] the studies focused on identifying tightly-connected clusters or communities within a given graph and inferring potential communities in a network based on density of linkage. In [17] the authors investigate an extensive characterization of the graph structure of the web with various features and consider the subgraph of the web consisting of all pages containing these features. In [18, 19] the self-identified nature of the online communities is used to study the relationship between different newsgroups on Usenet. In [20, 21, 22, 23, 24] social network evolution is studied as its members' attributes change.

The social network analysis and social clusters of the above work are mainly based on blogs, emails or the World Wide Web [23, 25, 26]. In [27, 28] the structure and tie strength of mobile telephone call graphs was investigated. In [29] the authors applied

the spectral clustering method to telephone call graph partition. In [30] the authors discovered the communities of mobile users on call detail records using a triangle approach. In [31] the algorithm based on clique percolation was developed to investigate the time dependence of overlapping social groups so as to discover relationships characterizing social group evolution and capturing the collaboration between colleagues and the calls between mobile phone users. In [32] the authors performed a new method for measuring human behavior, based on contextualized proximity and mobile phone data, to study the dyadic data using the nonparametric multiple regression quadratic assignment procedure (MRQAP), a standard technique to analyze social network data [33, 34], discover behavioral characteristics of friendship using factor analysis and predict satisfaction based on behavioral data. In [35] the authors studied the stability of social ties by defining and measuring the persistence of the ties. In [36] the authors propose a spreading activation-based technique to predict potential churners by examining the current set of churners and their underlying social network. In [37] the spatiotemporal anomalies of calls and patterns of calling activity are investigated using standard percolation theory tools. Almost all above research focused on large social networks and social groups. We focus on individual social groups using the probability model, *affinity*, which is different from the previous work on the measurements based on mobile phone call detail records.

3 Model

3.1 Formulation

Groups correspond to clusters of data. Cluster analysis concerns a set of multivariate methods for grouping data variables into clusters of similar elements.

In first step we apply affinity, that is measured in a probability scale, instead of simple/basic similarity coefficients. In the second step we define an aggregation criterion for merging similar clusters of elements. In the third step we use some way to assess the validity of the clustering results.

Affinity measures the similarity between probability measures. A related notion is the Hellinger distance. Since our problem belongs to discrete events, we only consider finite event spaces. Let

$$S_N = \{P = (p_1, p_2, \dots, p_N) \mid p_i \geq 0, \sum_{i=1}^N p_i = 1\}$$

be the set of all complete finite discrete probability distributions and $P, Q \in S_N$. The Hellinger distance between P and Q is defined as [4]

$$d_H^2(P, Q) = \frac{1}{2} \sum_{i=1}^N (\sqrt{p_i} - \sqrt{q_i})^2 \quad (1)$$

$$d_H^2(P, Q) \in [0, 1], \quad d_H^2(P, Q) = 0 \text{ if } P = Q \text{ and } d_H^2(P, Q) = 1 \text{ if } P \text{ and } Q \text{ are disjoint.}$$

The affinity between probability measures P and Q is defined as [4]

$$A(P, Q) = 1 - d_H^2(P, Q) = \sum_{i=1}^N \sqrt{p_i q_i} \quad (2)$$

$$A(P, Q) \in [0, 1], \quad A(P, Q) = 1 \text{ if } P = Q \text{ and } A(P, Q) = 0 \text{ if } P \text{ and } Q \text{ are disjoint.}$$

Proof.

$$\begin{aligned}
 A(P, Q) &= 1 - d_H^2(P, Q) = 1 - \frac{1}{2} \sum_{i=1}^N (\sqrt{p_i} - \sqrt{q_i})^2 = 1 - \frac{1}{2} \sum_{i=1}^N (p_i - 2\sqrt{p_i}\sqrt{q_i} + q_i) \\
 &= 1 - \frac{1}{2} \left(\sum_{i=1}^N p_i - 2 \sum_{i=1}^N \sqrt{p_i q_i} + \sum_{i=1}^N q_i \right) = \sum_{i=1}^N \sqrt{p_i q_i}
 \end{aligned}$$

For finite and discrete data, let $M(X, Y)$ be a $L \times N$ matrix, where X represents the set of data units and Y is a set of N categorical variables. In this paper γ_j ($j=1, \dots, N$) is a vector of frequencies. Thus γ_j may be represented by the L coordinates n_{ij} ($i=1, 2, \dots, L$) which is a frequency. We will refer to the j -th column profile as the corresponding conditional vector with $n_{ij} / \sum_{i=1}^L n_{ij}$. This profile vector may be a discrete conditional probability distribution law. It is often a profile or probability vector of the population, where the set X of L data unit represents a partition of some random sample of subjects in L classes. In this paper $p_i = n_{ij} / \sum_{i=1}^L n_{ij}$. The column profiles have a major role since the similarity between pairs of variables will be measured using an appropriate function, the affinity in this paper, of their profiles.

3.2 Real-Life Data Sets and Parameters

Real-life traffic profile: In this paper, the actual call logs are used for analysis. These actual call logs are collected at MIT [39] by the Reality Mining Project group for a period of 8 months. This group collected mobile phone usage of 100 users, including their user IDs (unique number representing a mobile phone user), time of calls, call direction (incoming and outgoing), incoming call description (missed, accepted), talk time, and tower IDs (location of phone users). These 100 phone users are students, professors and staff members. The collection of the call logs is followed by a survey of feedback from participating phone users for behavior patterns such as favorite hangout places; service providers; talk time minutes and phone users' friends, relatives and parents. We used this extensive dataset for our social group analysis in this paper. More information about the Reality Mining Project can be found in [39].

In our lives we have relationships with a small group of individuals in our social network such as family members, relatives, friends, neighbors and colleagues. Based on these social relationships, we divide the time of a day into working time (8am-5pm) and nonworking time (5:01pm-7:59am). Further, in the two time periods we divide our social network members into three categories: socially close members, socially near members and socially far members.

- *Socially Close Members:* These are the people with whom we maintain the strongest social relationship. Quantifying by phone calls we receive more calls from them and we tend to talk to them for longer periods of time. Family members, intimate friends and colleagues in the same team belong to this category.
- *Socially Near Members:* These relationships are not as strong as those of family members, intimate friends and colleagues in the same team. Sometimes, not always, we connect each other and talk for a considerably longer

periods. We mostly observe intermittent frequency of calls from these people. Distant relatives, general friends, colleagues in a different team and neighbors are in this category.

- *Socially Far Members*: These people have weaker relationships with each other in social life. They call each other with less frequency. We seldom receive calls from them and talk each other in short time.

We use the affinity formula (2) to classify social groups based on the time of the day, call frequencies, reciprocity and call duration.

Time of the day: Everyone has his/her own schedule for working, studying, entertainment, sleeping, traveling and so on. The schedule is mainly based on the time of the day and day of the week. We divide the time of the day into two parts: working time (8am-5pm) and nonworking time (5:01pm-7:59am).

Call frequencies: The call frequency is the number of incoming or outgoing calls in a period of time. The more the number of incoming or outgoing calls in a period of time, the more socially close the caller and callee relationship.

Call duration: The call duration is how long both caller and callee want to talk to each other. The longer the call duration is in a period of time, the more socially close the caller and callee relationship.

Reciprocity: Reciprocity represents the response by one party to calls from another party.

3.3 Computing the Affinity

In this paper, we use three attributes incoming (*in*), outgoing (*out*) and reciprocity (*reci*) of calls.

Let m_i, n_i be the number of calls, where $i \in \{in, out, reci\}$. $P = (p_{in}, p_{out}, p_{reci})$ is a vector of normalized frequencies over the training period and $Q = (q_{in}, q_{out}, q_{reci})$ is a vector of normalized frequencies of the same attributes observed over the testing period. Then

$$p_i = m_i / \sum_i m_i \text{ where } i \in \{in, out, reci\} \text{ and } q_i = n_i / \sum_i n_i \text{ where } i \in \{in, out, reci\}.$$

The affinity between P and Q is computed as follows:

$$A(P, Q) = \sum_i \sqrt{p_i q_i} \text{ where } i \in \{in, out, reci\} \quad (3)$$

We used the data from the data set of four months, the Fall semester since the communication members were relatively less changed in a semester for students. We compute the affinity values using formula (3) for four months of the call log data. We define:

- Socially close members if $0.7 < A(P, Q) \leq 1$
- Socially near members if $0.3 < A(P, Q) \leq 0.7$
- Socially far members if $0 \leq A(P, Q) \leq 0.3$

4 Experiment Results and Discussion

In Figure 1, the x-axis indicates the phone numbers that are used to communicate with user29 for four months, and the y-axis indicates the affinity values based on both number of calls and call duration respectively. From figure 1 user29 has seven socially close members, eight socially near members and twenty four socially far members in this four-month period. The details of group members are listed in table 1. In Table 1 we divide the social group members into work time members and non-work time members. In general, during work time we prefer to talk to colleagues, bosses, secretaries, clients and customers, occasionally speak to family members and friends for special cases, and during non-work time we usually talk to family members and friends, and we occasionally speak to colleagues, clients and customers for special cases. Note that some people may be both our work time colleagues and non-work time friends. Thus the set of work time members and the set of non-work time members may overlap. In Table 1 user29, who was a student, had one socially close member, two socially near members and one socially far member in work time and seven socially close members, eight socially near members and twenty-four socially far members in non-work time.

Figure 2 shows the call network of subset of call detail records in one month in which there are 326 vertices labeled by phone number ids which denote the communication members and the corresponding arcs representing the incoming or outgoing calls by the arrows. There are about 3200 communication members in the four month call detail records. Since the space is limited, we only use the call network of subset of call detail records to show the relationships among the communication members. The phone number id of user29 is 264 and the part of his communication members is shown in Figure 2.

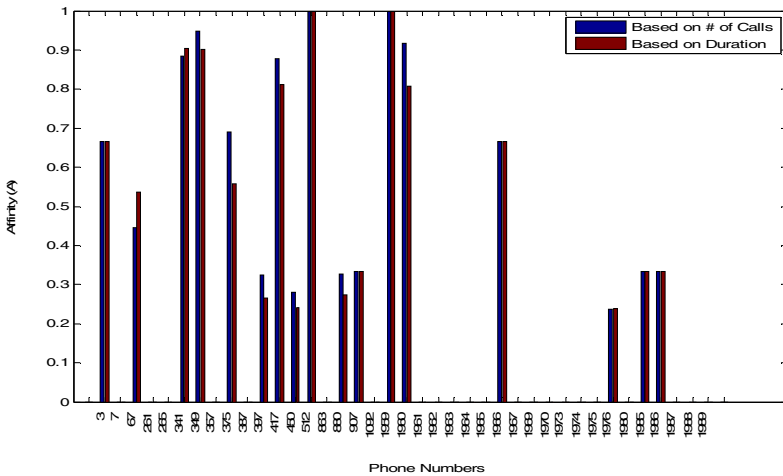


Fig. 1. The affinity values for user29

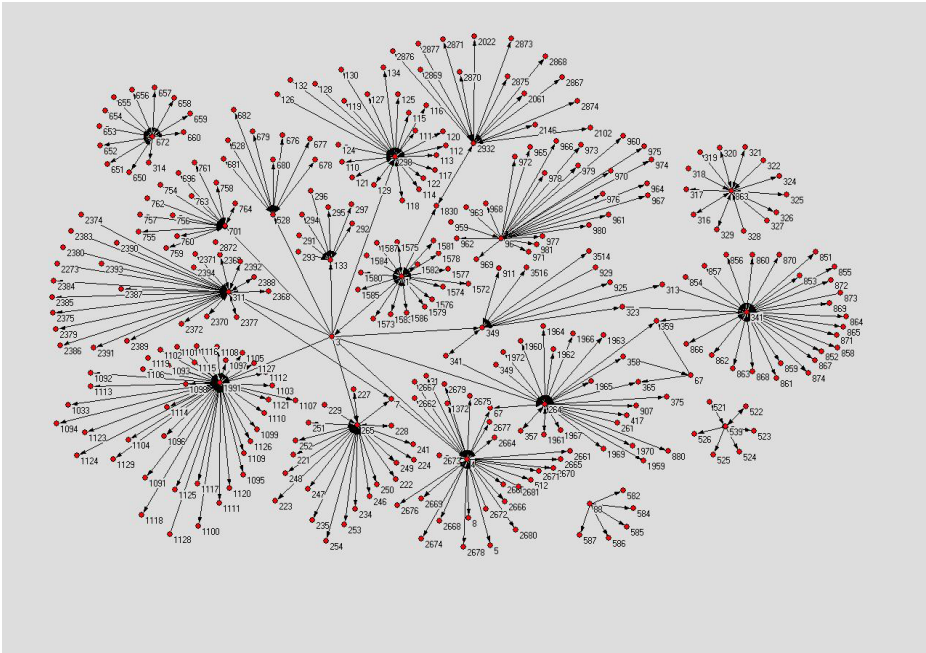


Fig. 2. The call network of subset of call detail records

Table 1. Social groups for phone user29 (The results in the Table 1 are computed by the equation 3)

User ID (Total # of members)	Work Time Member ID (# of members)			Non-work Time Member ID (# of members)		
	Close	Near	Far	Close	Near	Far
User29 (39)	1959 (1)	67, 1968 (2)	1967 (1)	265, 349, 375, 417, 512, 1959, 1960 (7)	3, 67, 397, 863, 907, 1966, 1985, 1986 (8)	7, 261,341, 357, 387, 417, 450, 863, 1092 1961,1962, 1963,1964, 1965,1969, 1970,1973, 1974,1975, 1976,1980, 1987,1988, 1989 (24)

5 Validation

To evaluate the accuracy of our model, we used actual call logs of 100 phone users and randomly choose 10 phone users. These users include students, professors and

staff members. The best way to validate the results is to contact the phone users to get feedback. But because of the privacy issues it is almost impossible to use this way. Thus we use quantitatively hand-labeling methods to validate our model. We have used the data of the four months to classify the social groups. In order to validate our model, we hand labeled the communication members based on the number of calls, duration of calls in the period, history of call logs, location, time of arrivals, and other humanly intelligible factors.

Table 2 describes the experimental results for 10 phone users. Our model achieves good performance with high accuracy of 94.19%.

Table 2. Social groups for phone users

User ID	Total # of members	Close	Near	Far	Hit	Fail	Unsure
29(student)	39	7	8	24	38	0	1
41(professor)	39	6	6	27	23	0	2
21(student)	20	5	2	13	18	1	1
74(student)	13	2	4	7	12	0	1
88(staff)	66	5	9	42	63	0	3
33(staff)	31	4	2	25	31	0	0
15(student)	29	10	4	15	25	2	2
49(student)	18	6	2	10	16	1	1
50(student)	63	6	14	43	61	0	2
95(professor)	8	1	4	3	8	0	0

6 Conclusion

In this paper we proposed the affinity model for classifying the social groups based on mobile phone call detail records. We use affinity to measure the similarity between probability distributions.

We may find the short-term friends, say a month, or long-term friends, say a year or more years using our model by adjusting the parameters.

This work is useful for enhancing homeland security, detecting unwanted calls (e.g., spam), communication presence, marketing etc. The experimental results show that our model achieves good performance with high accuracy.

In our future work we plan to detail the social group classification, analyze the social group evolution and study the social group dynamics.

Acknowledgement

We would like to thank Nathan Eagle and Massachusetts Institute of Technology for providing us the call logs of Reality Mining dataset.

This work is supported by the National Science Foundation under grants CNS-0627754, CNS-0516807, CNS-0619871 and CNS-0551694. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Wasserman, S., Faust, K.: *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge (1994)
2. Brass, D.J.: A social network perspective on human resources management. *Research in Personnel and Human Resources Management* 13, 39–79 (1995)
3. Carley, K.M., Lee, J., Krackhardt, D.: Destabilizing networks. *Connections* 24(3), 79–92 (2002)
4. Ogatha, H.: Computer Supported Social Networking for Augmenting Cooperation. In: *Computer Supported Cooperative Work*, vol. 10, pp. 189–209. Kluwer Academic Publishers, Dordrecht (2001)
5. Donath, J., Karahalios, K., Viegas, F.: Visualizing conversation. In: *Proceeding of Hawaii International Conference on System Sciences*, vol. 32 (1999)
6. Sack, W.: Conversation Map: A text-based Usenet Newsgroup Browser. In: *Proceeding of ACM Conference on Intelligent User Interfaces*, pp. 233–240 (2000)
7. N.N. Microsoft Netscan, <http://netscan.research.microsoft.com>
8. Brickley, D., Miller, L.: FOAF Vocabulary Specification, Namespace Document (2004), <http://xmlns.com/foaf/0.1>
9. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. In: *Proceedings of the National Academy of Sciences of United State of America*, vol. 99(12), pp. 7821–7826 (2002)
10. Newman, M.E.J.: Modularity and community structure in networks. In: *Proceedings of the National Academy of Sciences*, vol. 103, pp. 8577–8583 (2006)
11. Broder, Kumar, A.R., Maghoul, F., Raghavan, Rajagopalan, P.S., Stata, R., Tomkins, A., Wiener, J.: Graph structure in the web. *Computer Networks* 33(2), 309–320 (2000)
12. Doreian, P., Batageli, V., Ferligoj, A.: *Generalized Blockmodeling*. In: Granovetter, M. (ed.), Cambridge University Press, Cambridge (2005)
13. Flake, G., Lawrence, S., Giles, C.L., Coetzee, F.: Self-Organization and Identification of Web Communities. *IEEE Computer* 35(3) (2002)
14. Flake, G.W., Tarjan, R.E., Tsioutsouluklis, K.: Graph Clustering and Minimum Cut Trees. *Internet Math*. vol. 1 (2004)
15. Hopcroft, J., Khan, O., Kulis, B., Selman, B.: Natural communities in large linked networks. In: *Proceeding of 9th SIGKDD* (2003)
16. Newman, M.E.J.: Detecting community structure in networks. *Eur. Phys. J. B* 38, 321–330 (2004)
17. Dill, S., Kumar, R., McCurley, K., Rajagopalan, S., Sivakumar, D., Tomkins, A.: Self-similarity in the Web. In: *27th International Conference on Very Large Data Bases* (2001)
18. Borgs, C., Chayes, J., Mahdian, M., Saberi, A.: Exploring the community structure of newsgroups. In: *Proceeding of 10th ACM SIGKDD* (2004)
19. Viegas, F., Smith, M.: Newsgroup Crowds and AuthorLines. In: *Hawaii International Conference on System Science* (2004)
20. Holme, P., Newman, M.: Nonequilibrium phase transition in the coevolution of networks and opinions. *arXiv physics/0603023* (2006)
21. Sarkar, P., Moore, A.: *Dynamic Social Network Analysis using Latent Space Models*. SIGKDD Explorations: Special Edition on Link Mining (2005)
22. Backstrom, L., Huttenlocher, D., Kleinberg, J.: Group formation in large social networks: membership, growth, and evolution. In: *Proceedings of the 12th ACM SIGKDD*, pp. 544–554 (2006)

23. Kossinets, G., Watts, D.: Empirical analysis of an evolving social network. *Science* 311, 88–90 (2006)
24. Wang, X., McCallum, A.: Topics over Time: A Non-Markov Continuous-Time Model of Topical Trends. In: *Proceeding of 12th ACM SIGKDD* (2006)
25. Kumar, R., Novak, J., Raghavan, O., Tomkins, A.: Structure and Evolution of blogspace. *Communications of ACM* 47(12), 35–39 (2004)
26. Kumar, R., Novak, J., Tomkins, A.: Structure and Evolution of on line social networks. In: *Proceedings of the 12th ACM SIGKDD* (2006)
27. Nanavati, A.A., Gurumurthy, S., Das, G., Chakraborty, D., Dasgupta, K., Mukherjea, S., Joshi, A.: On the structural properties of massive telecom graphs: Findings and implications. In: *Proceedings of the Fifteenth ACM CIKM Conference* (2006)
28. Onnela, J.P., Saramaki, J., Hyvonen, J., Szabo, G., Lazer, D., Kaski, K., Kertesz, J., Barabasi, A.L.: Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences of United State of America* 104(18), 7332–7336 (2007)
29. Kurucz, M., Benczur, A., Csalogany, K., Lukacs, L.: Spectral Clustering in Telephone Call Graphs. In: *Proceedings of the Joint 9th WEBKDD and 1st SNA-KDD Workshop* (2007)
30. Teng, W., Chou, M.: Mining communities of acquainted mobile users on call detail records. In: *Proceedings of the 22nd Annual ACM Symposium on Applied Computing* (2007)
31. Palla, G., Barabasi, A., Vicsek, T.: Quantifying social group evolution. *Nature* 446, 664–667 (2007)
32. Eagle, N., Pentland, A., Lazer, D.: Inferring Social Network Structure using Mobile Phone Data. *Science* (in submission), http://reality.media.mit.edu/pdfs/network_structure.pdf
33. Baker, F.B., Hubert, L.J.: The analysis of social interaction data. *Social Methods Res.* 9, 339–361 (1981)
34. Krackhardt, D.: Predicting With Networks - Nonparametric Multiple-Regression Analysis of Dyadic Data. *Social Networks* 10(4), 359–381 (1988)
35. Hidalgo, A.C., Rodriguez-Sickert, C.: The Dynamics of a Mobile Phone Network. *Physica A* 387, 3017–3024 (2008)
36. Dasgupta, K., Singh, R., Viswanathan, B., Chakraborty, D., Mukherjea, S., Nanavati, A.: Social Ties and their Relevance to Churn in Mobile Telecom Networks. In: *Proceedings of the 11th ACM international conference on Extending database technology: Advances in database technology* (2008)
37. Candia, J., Gonzalez, M.C., Wang, P., Schoenharl, T., Madey, G., Barabasi, A.: Uncovering individual and collective human dynamics from mobile phone records. *Journal of Physics A: Mathematical and Theoretical* 41, 224015 (2008)
38. Fannes, M., Spincemaile, P.: The mutual affinity of random measures. *Periodica Mathematica Hungarica* 47(1-2), 51–71 (2003)
39. Massachusetts Institute of Technology: Reality Mining (2008), <http://reality.media.mit.edu/>
40. Zhang, H., Dantu, R.: Quantifying the presence for phone users. In: *Proceeding of Fifth IEEE Consumer Communications & Networking Conference* (2008)

PALASS: A Portable Application for a Location-Aware Social System

Martin Press, Daniel Goodwin, and Roberto A. Flores

Christopher Newport University,
Department of Physics, Computer Science & Engineering
1 University Place, Newport News, VA, USA 23606
{mpress, dgoodwin, flores}@pcs.cnu.edu,
<http://www.pcs.cnu.edu>

Abstract. In this paper we present PALASS, a location-aware framework for portable handheld devices based on the Google Android platform. Location-aware systems are context-sensitive systems that provide information based on current location. We expand this model to include a social dimension where events are shared among users of social groups and are graphically displayed using maps in Android handheld devices. The framework is built as a web service supporting the dynamic addition of geographically-located social events, which are automatically filtered according to the group affiliations of each user. To emphasize the implementation-independence of this approach, the framework defines roles for different functionalities, each with clearly defined tasks that simplify the overall interface of the system. Other location-aware systems are also explored to show how they relate to the PALASS framework.

Keywords: Location-aware systems, Handheld devices, Google Android, Web services, Social events, Social groups.

1 Introduction

In today's modern world, portable handheld communication devices have become abundant and an easily accessible way to exchange individual and collective information. New enhancements, such as web access, graphical user interfaces, and built-in geographical positioning systems (GPS) have advanced the application and functionality of these mobile devices. One of the most recent developments in this area is the Google Android platform [15], which is a Linux-based system that supports the development of Java applications for mobile devices, and allows the seamless integration of these applications with Google Maps. These characteristics make Android a suitable platform to develop a location-aware social system for mobile devices in which users visualize group event information as geographically-arranged symbols in maps.

A location-aware system is a context-sensitive system that is able to provide information based on current location. We follow the definition in [13] and consider a mobile device as an independent device capable of information management and

communication. There are two predominant technologies supporting the calculation of positioning in these devices: Wi-Fi and GPS, both of which have been used by location-aware systems in the past (in [5][12] and [9][11] respectively). On the one hand, Wi-Fi determines location by detecting differences in signal strength and direction and is used primarily for smaller indoor areas. On the other hand, GPS determines location by triangulating satellite signals and is primarily used outdoors because of signal interference within buildings.

In this paper we present a framework for location-aware systems that we implemented in Google Android and which we named PALASS (Portable Application for a Location-Aware Social System). Section 2 gives an overview of the social context for the system. The details of the PALASS framework can be found in section 3, and a use case scenario is presented in section 4. Lastly, a review of related work, and future improvements and conclusions are located in sections 5 and 6 respectively.

2 Social Context

In this section we present a few social concepts fundamental to our system.

A *place* is defined as the events done at a location by a person, where a *location* is a geographical position and an *event* is a happening that must be bound to a location and to *time constraints* [7]. An event that has these two criteria is considered to be *performable*. Events that are missing either of these criteria are considered incomplete and cannot exist in a location-aware system.

In a community-centered location-aware system, groups exist to promote social interaction between users. A *social group* can be defined as a collection of people who have a sense of common identity and frequently interact with each other [3]. Users with similar interests can join and create groups with specific activities. This is similar to the groups used on web based social networking sites such as *Facebook* and *MySpace*. Unlike an event, a group does not require a time and a location to exist. A group can have events at specific locations and at certain times, but that does not mean that the group is bound specifically to that location and time. For example, a university club holds a meeting in a classroom at 6 pm on a Monday. If the meeting is cancelled or postponed, this event is changed, but the group itself is still in existence. A group's existence is based on its membership. As long as there is at least one member of a group, the group still exists. Just as groups can have multiple events at various times, an individual can be a member of multiple groups at the same time. Many people have widely varied interest, and therefore can belong to different social groups with similar or different areas of interest. Using social groups is a step forward in creating a social network system [16].

Now that the basic terminology for social groups and events has been introduced, it is necessary to address the issues in creating a location-aware system and how these concepts fit into its design. These issues include: the distinction between places, user willingness to share information, and types of information.

2.1 Place and Location Distinction

One challenge is distinguishing between places and locations. For any useful information about a place to be displayed, one must be able to distinguish between events. One example of a location that represents two different places is a restaurant. For a chef, the restaurant is a place of work and useful information would be for him/her whether food is running low, allowing time to get more supplies. For a customer; however, the restaurant is a place of service. The daily special may be a piece of information deemed useful. One way to achieve this versatility is to allow the user to configure the device for each location, which is a tedious process that would require the user to manually select each option deemed useful. One solution is suggested by Kaasinen [8], who suggests that adaptive systems can be created by using group profiles to filter collective information. Using an adaptive system would greatly unclutter the user interface and since mobile devices have a small display, would make it an ideal solution.

2.2 Information Modeling

Another challenge is to identify useful information about a place to a specific user. According to the People-to-People-Geographical Place Study (P3 Study) [7], there are two types of information that can be given about a location.

The first type is static information, which does not change in a timely fashion. An example of this could be a menu for a well-established restaurant. It was found that this type of information is mostly useful to people who have not been to or rarely visit a location. Using this principle, static information should only be notified or displayed to users if they request it. Since static information is not bound by time it can only be bound to a location and not a place.

The second type is dynamic information, which is non-predictable and is bound by time constraints. This type of information has a high degree of usefulness to the user, who should be alerted of its occurrence. One example of this would be if a restaurant caught on fire, then the owner and employees would need to be aware of this event as soon as possible.¹

In addition to these, Kassinen [8] addresses two types of information that could be useful about a location. The first form is information generated by businesses in the form of advertising. As he pointed out, the usefulness of location-aware advertising varies based on age groups, where the younger demographic found it more useful. However, excessive use was found to be an annoyance due to a constant bombardment of unsolicited information. The second type of information that was highly rated by users was the information that other users have posted at a location.

¹ Although a fire is bound by certain time constraints, the duration of such an event cannot be predicted and therefore its ending time is not known until its completion. Still, events such as this one can be created before the complete duration is known.

2.3 Information Sharing

The willingness to share data is a key issue of any system that provides community interaction. There are three levels of sharing of data for an application: personal, social, and urban [10]. The personal level deals with data that one would not want others to see, such as a bank statement. The next level is social sharing where a controlled group is allowed access. Lastly, urban sharing allows access by anyone within the system.

Problems can occur if a social system does not have enough levels of information sharing or if information cannot be easily migrated between levels. Results from the Place Mail Study [9], which analyzed user willingness to share messages, showed that people prefer having control over who can view messages. Users naturally chose private over global viewing of messages, and mentioned that having a buddy list would allow messages to many recipients while keeping control on them. One problem detected was that the type of sharing a user wants on a particular piece of information may change over time; for instance, a man may not want to share his position when taking the train in the morning but may be willing in the afternoon, when he is open to share this information with friends and coworkers. It can be explained that he was rushed in the morning and was not in a good mood to meet others. This shows that the level of sharing may directly relate to an event and may vary for every user.

To facilitate the handling of information sharing, we used social groups as an abstraction for different levels. Personal sharing can be achieved by only allowing a single member in a group. Social sharing can be achieved by allowing individuals with common interests to join. Lastly, urban level of sharing is a group in which all members of a system belong. Using this abstraction eliminates the need for a separate hierarchy for levels of information sharing.

3 Framework Overview

The PALASS framework is modeled as a web service that allows the dynamic sharing of social events among handheld devices using geographical location. Users use a subscribe mechanism to join groups, and receive events shared between users of the group.

As shown in Figure 1, the framework is divided into a client side and a server side, with client and server applications conforming to particular roles. The server side provides a location-aware web service that receives new events and provides existing events to users (*capturing* and *provider* roles respectively). On the other hand, the client side has applications that generate new events and consume existing event information (*generator* and *consumer* roles respectively). These roles are described in the sections below.

3.1 Client Roles

The framework contains two types of client roles: a *generator* (which has the functionality required to input a new event into the system) and a *consumer*

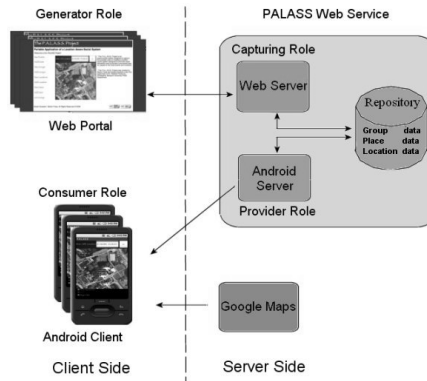


Fig. 1. The PALASS framework

(which supports the representation of events in location-aware devices). Although nothing precludes one application to perform these roles at the same time, we separated their implementation due to the innate limitations in the user interface of mobile devices. In particular, mobile devices have limited output (small screen) and input (non-ergonomic keyboard) capabilities. As noted in [8,16], the features of handheld devices make them more suitable to instant notification than to focal points of attention for prolonged periods of time.

In our PALASS implementation, the *generator* role is fulfilled by a web portal, and the *consumer* role by Google Android mobile devices. Using the web portal users can create new social groups, locations, events, and perform membership changes. Google Android clients fulfill *consumer* roles in which they request events to a provider (server) application based on their current location and user profile. All message information exchanged between consumer and provider applications are done using XML messages.

Figure 2 shows the three narrowing levels of information displayed in our client implementation, to which (borrowing Android's terminology) we refer to as *activities*. These activities, which are the *MapView*, *ListView* and *ItemView*, allow users to traverse between increasingly specific event information.

The *MapView* activity displays the Google Map region in which the device is located.² Given the user profile communicated to the server, the *MapView* activity only displays events of groups that the user is a member of. To further simplify, events at a location are displayed using a call-out shape containing colored dots, where each dot indicates the existence of at least one event for the group represented by that color. The lower part of the screen shows a list of groups, each with its corresponding color, thus allowing users to associate groups with colored dots in the call-outs displayed. As shown in the figure, this activity can display as many call-out as locations with events exist in the map displayed.

² In addition to using the built-in GPS as input for the map coordinates to display, our implementation allows users to scroll to surrounding areas on the map or to provide a latitude-longitude coordinate for the location to display.

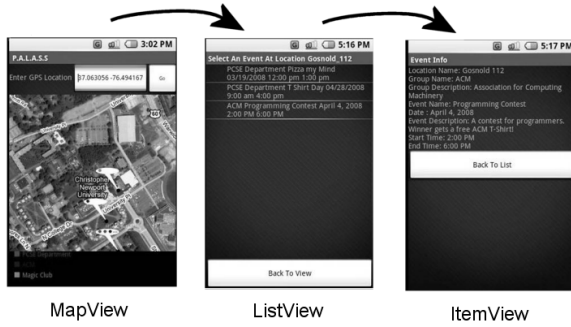


Fig. 2. Android client activities

Clicking on a call-out displays the *ListView* activity, which lists all events available at that location. The maximum groups visible at one time is nine, which is between the 7 ± 2 interval of Miller's law. To increase readability, this activity only lists the name and time of events, and the group to which each belongs.

Lastly, clicking on an event in the *ListView* activity leads to the *ItemView* activity, which displays a detailed description of the selected event.

3.2 Server Roles

The framework has two types of server roles: a *capturing* role (which records new events provided by generator clients) and a *provider* (which supplies filtered events to consumer clients based on user profiles).

To implement the *capturing* role, we used a web server application that receives information from the web portal for storage in a database repository, including information of events, locations, users and groups. In the case of users, the system implements an urban group to which all users are automatically added and to which all users have access by default.

We implemented the *provider* role using a multithreaded server written in Java. The responsibility of this server is to provide place, group, and location information based on the context of the Android client, which includes a user identification and its geographical position. As previously mentioned, the client *consumer* and server *provider* communicate exclusively through XML messages.

The information manipulated by the *capturing* web and *provider* Java server applications is stored in a MySQL database repository. For simplicity, and to ensure data integrity across all clients in a simple manner, our current prototype does not allow the deletion of data from the database. To overcome this limitation, we foresee extending the prototype to allow the synchronization of information either at fixed intervals, upon request, or immediately after changes relevant to the user context are detected.

Lastly, Google Maps [4] is the final component belonging to the server side shown in Figure 1. Although not a direct component of our framework, Google Maps is accessed transparently by Android clients in the *MapView* activity. In brief, the Google Maps Server is an image server that sends scaled images back to Android clients based on a provided GPS coordinate. Embedded in these images are the bounds for the latitude and longitude of the image sent.

3.3 Technical Details

Although PALASS was developed in Java, there were notable differences between JDK and Google Android's SDK. For example, Android uses its own classes instead of those in Swing/AWT to handle GUI components, which are created using XML layout files read at runtime. This neatly separates GUI functionality and design. Also, Android integrates Google Maps API smoothly into its SDK. It allows loading scalable images based on GPS locations, drawing on map surfaces, and doing GPS-to-pixel calculations. We used a *TimerTask* object to synchronize maps and GPS movements, and noticed that the bytecode produced by Android is proprietary and runs on a custom virtual machine.

In addition, we designed PALASS to use patterns promoting a more modular design. These include the observer pattern for drawing clickable callout locations; the singleton pattern to support inter-activity communication; and the factory pattern for text label instantiation.

4 Use Case Scenario

The restaurant scenario is an example of how a location-aware system could be beneficial to users. In this case, suppose a chef arrives at his restaurant and finds that the owner has left a note stating that some of the ingredients for the house special were running low. The chef promptly gets into his car and drives to the nearest supermarket in search of the needed ingredients. However, the store has sold out some of the needed items. Meanwhile patrons at the restaurant have requested the house special. Unfortunately the restaurant could only create a few orders before running out of the needed ingredients. After the chef finally locates all of the ingredients and returns to the restaurant several patrons had become frustrated by the long wait and had left.

Using the restaurant use case, the following scenario shows how PALASS would function in the real world. The chef on duty at the restaurant realizes that the supplies for the daily special are running low. He goes into the office at the restaurant and uses the PALASS Web Portal to add an event to his restaurant employees group. At this time, the chef for next shift is getting ready for work and notices that a new event has been posted. After reading the event, he decides to stop and pick-up the item on the way to work to save time. He checks events for the local supermarkets group and discovers that a nearby supermarket has some of the needed items on sale. The chef stops by the supermarket on the way to work, picks up the ingredients, and arrives at work in time for the noon rush.

5 Related Work

In recent years, researchers have explored the viability of location-aware systems in daily life situations. Below we list several systems from we sought inspiration to define our PALASS framework.

Place Mail [9] is a location-aware system that allowed private text based messages to be stored at various geographical locations. Users would be alerted when reaching a location in which a reminder had been stored, and would be able to view previously posted reminders to discover the social meaning of a place. The system featured a map, and could find previously recorded messages by using a web portal called *Sharescape*. As noted in the study, one limitation of the system was that users found too time consuming to use their devices to indicate which messages to share. The PALASS framework minimizes this issue by implementing social groups thus eliminating the need for configuring sharing on a message-by-message basis.

Another location-aware system we identified was E- Graffiti [1]. Unlike Place Mail, E-Graffiti allowed the posting of messages to a particular person. It also included the ability to save private and global messages at a location. The system included a graphical picture of the current location, such as a building where a location was identified. The authors noted that users were confused when using the system, and would post messages at improper locations believing anyone could view them. However, only participants at the current location of the posted message could view them. To curtail this issue, the PALASS system incorporates a Google Map-based display system that allows users to explore the environment virtually without needing to be at the exact location.

Active Campus [5] is a location-aware system that included graphical features, such as a map that moved with the user. It would identify buddies and location information. The system incorporated both Wi-Fi location detection, as well as GPS for outdoor areas where a dense Wi-Fi network was not available. The client for the Active Campus System was lightweight and consisted of a web page and one other application to detect the user's geographical position. One problem noted by the authors was the scalability of the system. The client's web page required a proprietary image server that must be maintained. As the system grew, new servers would need to be added. The PALASS system uses Google Maps to minimize this issue.

Another location-aware system that was developed is the location-aware event planner described in [11]. It incorporates a moving map and user interface to track people and events. Users are able to create places and set their own visibility level. The visibility level lets users decide who can view them at any given time. It also defines social groups as a way of creating events. The authors noted during the initial evaluation of the prototype that the portable keyboard used to perform various tasks was uncomfortable to some participants. The PALASS framework removes the complication of entering text for creating events and social groups by moving this role functionality to a web portal, thus simplifying the clients use.

6 Conclusions and Future Work

Even with the PALASS framework's large feature set it still has room for improvement in both client and server implementations.

Currently there is no automated notification system for events other than a new object appearing on the screen. For instance, if a new event is added to the system, the user will not be notified of the event unless the user is viewing the screen. A possible solution would be to make the mobile device vibrate or ring when a new event is added. Although notifying users of the arrival of new events in real time (regardless of whether or not the user is able to constantly monitor the screen) could be an enhancement to the system, Kaasinen [8] noted that unsolicited notifications could potentially become an inconvenience for users. Because of this, notification features would require a filtering mechanism with the ability to be turned off completely, which would allow user to use other applications undisturbed while still been notified of new events.

Another possible improvement to the Android client is to identify the location of members to social groups that the user is associated with, in a similar fashion as it is done in the Active Campus system [5]. This would further help to promote social interaction by letting users know the location of other members of the group. It would also let users know a how many members of a group are at a particular location. This could be beneficial if the user wanted to know population based information.

The web portal could also be improved by adding a clickable map to pinpoint locations, allowing the visual addition of new locations to the system. Another future milestone will be transitioning our prototype to a sizeable user base once Android phones become available. As mentioned by Gerding and Ehrlich, a critical mass of users is required for the success of social networking software [6,2]. A complementing way to further this goal is to incorporate PALASS into existing social networking websites, such as Facebook, where we could take advantage of the large number of already created groups.

As the capabilities and functionality of mobile phones improve and evolve over time, their areas of application expand toward new uncharted limits. Implementing technology to make these mobile devices aware of their current location has the potential to open a whole new world of applications for mobile devices. The PALASS, in its current implementation and with the proposed future improvements, offers a number of practical functionalities using this technology. It promotes social interaction between people with similar interests, allows users to conveniently search for local events, and allows users to be notified when new events become available. With the rich feature set of PALASS and possible expansions, it can be a unique tool for the digital age.

Acknowledgements

We are thankful to the anonymous reviewers for their helpful insight and suggestions. Furthermore, we express gratitude to the department of Physics, Computer Science and Engineering and Christopher Newport University for their support.

References

1. Burrell, J., Gay, G.K.: E-Graffiti: Evaluating real-world use of a context-aware system. *Interacting with Computers*, Elsevier Science 14(4), 301–312 (2002)
2. Ehrlich, S.F.: Strategies for encouraging successful adoption of office communication systems. *ACM Transactions on Office Information Systems* 5(4), 340–357 (1987)
3. Giddens, A., Dunier, M., Appelbaum, R.: *Introduction to Sociology*. W.W. Norton & Company (2005), <http://www.wwnorton.com/college/soc/giddens5/>
4. Google Maps API, <http://code.google.com/apis/maps/documentation/events.html>
5. Griswold, W., Boyer, R., Brown, S.W., Truong, T., Bhasker, E., Jay, G., Shapiro, R.B.: Active Campus - sustaining educational communities through mobile technology. Technical Report CS2002-0714, UC San Diego, Department of CSE (2004)
6. Grudin, J.: Groupware and social dynamics: eight challenges for developers. *Communications of the ACM* 37(1), 92–105 (1994)
7. Jones, Q., Grandhi, A., Whittaker, S., Chivakula, K., Terveen, L.: Putting systems into place: a qualitative study of design requirements for location-aware community systems. In: *CSCW 2004. Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pp. 202–211. ACM Press, New York (2004)
8. Kaasinen, E.: User needs for location-aware mobile services. *Personal Ubiquitous Computing*, Springer-Verlag 7(1), 70–79 (2003)
9. Ludford, P., Priedhorsky, R., Reily, K., Terveen, L.: Capturing, sharing, and using local place information. In: *Conference on Human Factors in Computing Systems, Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 1235–1244 (2007)
10. Parker, A., Reddy, S., Schmid, T., Chang, K., Saurabh, G., Srivastava, M., Hansen, M., Burke, J., Estrin, D., Allman, M., Paxson, V.: Network system challenges in selective sharing and verification for personal social and urban scale sensing applications. In: *Proceedings of the Fifth Workshop on Hot Topics in Networks (HotNets-V)*, pp. 37–42 (2006)
11. Pousman, Z., Iachello, G., Fithian, R., Moghazy, J., Stasko, J.: Design iterations for a location-aware event planner. *Personal Ubiquitous Computing* 8(2), 117–125 (2004)
12. Rerrer, U.: Location-aware web service architecture using WLAN positioning. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM-WS 2005*. LNCS, vol. 3762, pp. 196–205. Springer, Heidelberg (2005)
13. Sandoval, G.L., Chavez, E.E., Caballero, J.C.P.: A development platform and execution environment for mobile applications. *CLEI Electronic Journal* 7(1), Paper 4 (2004)
14. Schwinger, W., Grün, C., Pröll, B., Retschitzegger, W.: A Light-Weight framework for location-based services. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM-WS 2005*. LNCS, vol. 3762, pp. 206–210. Springer, Heidelberg (2005)
15. What is Android? <http://code.google.com/android/what-is-android.html>
16. Wojciechowski, A.: Supporting social networks by event driven mobile notification services. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM-WS 2007, Part I*. LNCS, vol. 4805, pp. 398–406. Springer, Heidelberg (2007)

Toward the Development of an Integrative Framework for Multimodal Dialogue Processing

Arianna D'Ulizia, Fernando Ferri, and Patrizia Grifoni

IRPPS-CNR, via Nizza 128, 00198 Roma, Italy

{arianna.dulizia, fernando.ferri, patrizia.grifoni}@irpps.cnr.it

Abstract. The “universal accessibility” concept is acquiring an important role in the research area of human-computer interaction (HCI). This phenomenon is guided by the need to simplify the access to technological devices, such as mobile phones, PDAs and portable PCs, by making human-computer interaction more similar to human-human communication. In this direction, multimodal interaction has emerged as a new paradigm of human-computer interaction, which advances the implementation of universal accessibility. The main challenge of multimodal interaction, that is also the main topic of this paper, lies in developing a framework that is able to acquire information derived from whatever input modalities, to give these inputs an appropriate representation with a common meaning, to integrate these individual representations into a joint semantic interpretation, and to understand which is the better way to react to the interpreted multimodal sentence by activating the appropriate output devices. A detailed description of this framework and its functionalities will be given in this paper, along with some preliminary application details.

Keywords: Multimodal languages, System Architecture, Human-Computer Interaction.

1 Introduction

In recent years, people are increasingly surrounded by objects in the everyday environment that are equipped with embedded software and wireless communication facilities. For instance, mobile phones, PDAs and laptops are used by an increasing amount of people to carry out everyday activities. This phenomenon produces the need to simplify the access to these technological devices making human-computer interaction more similar to human-human communication. As a consequence, the “universal accessibility” concept is acquiring an important role in the research area of human-computer interaction (HCI). Three of the main emerging research directions of the HCI, in line with the universal accessibility concept, are: (i) to make this interaction more intuitive, natural and efficient by integrating multiple input-output modalities, (ii) to enable a broader spectrum of users, with different ages and skill levels as well as users with disabilities, to access technological devices, and (iii) to increase the level of freedom offered to users. In particular, multimodal interaction, which refers to the simultaneous or alternative use of several modalities, has emerged as the paradigm of human-computer interaction for the implementation of universal accessibility.

This paper, according to the universal accessibility concept, proposes the architectural framework collecting the general features of multimodal systems, used to develop a multimodal dialogue processing platform. Section 2 provides a short description of the multimodal communication process characterizing some of the multimodal systems more discussed in the literature. Section 3 introduces the architectural features of the proposed platform. Finally section 4 details the two main modules of the multimodal dialogue processing platform (the multimodal language interpreter and the modelling components), and gives a description of the application scenario of monitoring patients with neurodegenerative diseases. Section 5 concludes the paper, providing some perspectives for future works.

2 Background

The success of the human-computer communication depends on the reaching of a common ground by exchanging information through the communication modalities. Such a communication modality refers to the medium or channel of communication that conveys information [1]. Multimodality refers to the “quality” of a system to allow more than one communication modality to be used during human-computer interaction. A general model of multimodal human-computer communication is shown in Figure 1. It is composed of four main input/output components, according to the study of Schomaker et al. [2]:

- the *human output modalities*, that are devoted to control and manipulate computational systems by achieving a high level of interactivity and naturalness of the multimodal interface. The speech is the dominant modality that carries most of the information content of a multimodal dialogue. However, gesture and gaze modalities are extensively studied in literature as efficient input modalities that are better suited to represent spatio-temporal information and are usually complementary modalities of the speech input;
- the *human input channels*, that are devoted to perceive and acquire information coming from the feedback channels of computational systems. The most frequently used perception channels are eyes, ears and touch, among which the first is the dominant input modality that receives the most information flow, followed by the auditive and tactile channels;
- the *computer input channels*, through which the computer gets information from the human output modalities. Some examples of computer input channels are microphone, camera, keyboard, mouse. Once acquired, the inputs need to be brought together and interpreted in order to give a coherent meaning to the multimodal act of the user;
- the *computer output modalities*, that are devoted to give feedback to the user, as, for instance, visual feedback, speech synthesizer, haptic feedback and so on.

So that the multimodal human-computer communication process takes place successfully, the actions that the user expresses through the human output modalities have to be acquired by the system through the computer input modalities, and the

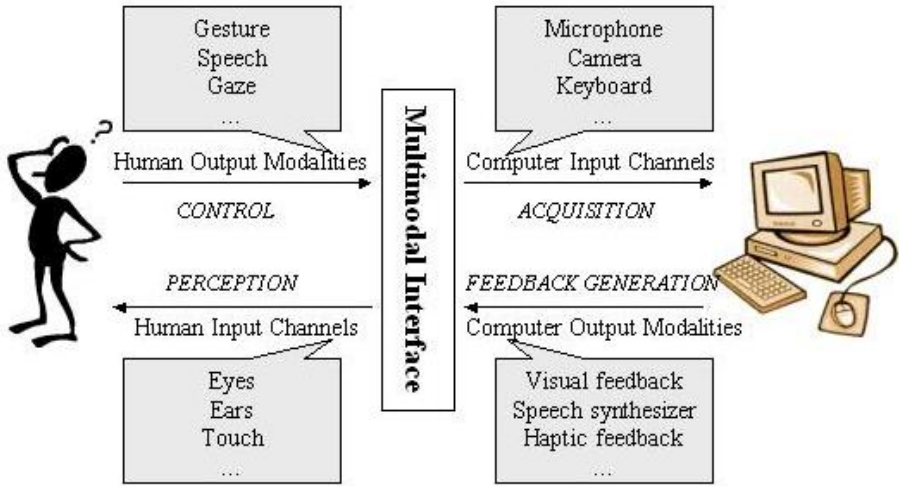


Fig. 1. The multimodal human-computer communication process

human input channels of the user have to be able to perceive and understand feedback from the computer output channels.

The informational flow that involves the human output and computer input modalities is named *input flow*, whereas the flow that involves the human input and computer output channels is named *feedback flow*.

Multimodal systems have been largely studied since the 80's when the first original system "put-that-there" was developed by Bolt [3]. This system used the speech and the location of a cursor on a touchpad display to allow simple deictic reference, as for example "create a blue square here". As well as the "put-that-there" system, several attempts to overcome common graphical user interface have been made since the 90's until now [4] [5] [6] [7] [8]. CUBRICON [4] used typed and spoken sentences and deictic mouse clicks as input in order to interact with a two-dimensional map. MATIS (Multimodal Airline Travel Information System) [5] allows the user to ask for information about the air flights departure/arrival time by using speech and pen-based gesture modalities, along with mouse clicks and keyboarding. QuickSet [6] was developed with the aim of training Californian military troops and used speech and pen-based gestures to interact with a geo-referenced map. QuickTour [7] is a multimodal system that enables a spoken and pen-based interaction to navigate geographical maps. Smartkom [8] is another multimodal dialogue system that merges gesture, speech and facial expressions for both input and output via an anthropomorphic and affective user interface.

The attention on the multimodal human machine communication approach has been actually focused on the architectures of context-aware multimodal systems for human machine interaction. According to this perspective, the following section proposes the architectural framework collecting the general features of multimodal systems.

3 Architectural Requirements

From the analysis of current multimodal systems, we envisaged that the challenge of the future multimodal systems is to move toward frameworks able to manage multimodal communication between people and the environment in different application scenarios. Toward this goal, we developed a multimodal dialogue processing platform that is able to acquire information derived from whatever input modalities, to give these inputs an appropriate representation with a common meaning, to integrate these individual representations into a joint semantic interpretation, and to understand which is the better way to react to the interpreted multimodal sentence by activating the appropriate output devices. The architecture of this platform is depicted in Figure 2.

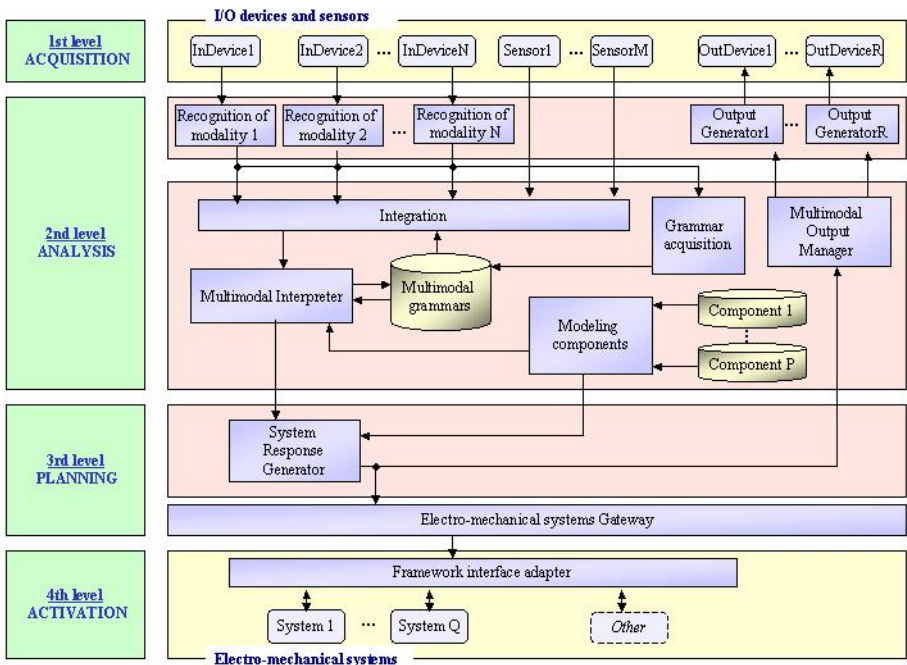


Fig. 2. Architecture of the multimodal dialogue processing platform

The individual components of the architecture are specified in terms of their tasks, responsibilities and dependencies that are necessary to provide an integrative, configurable, scalable, and adaptive framework for human-machine communication. In order to realize a more robust, natural and flexible integration of human signals in an efficient and unambiguous way, four different architectural levels are envisaged:

- *The acquisition level:* This level includes the specific I/O devices, such as, for example, display, cameras, microphone and loudspeakers, and input sensors.

- *The analysis level:* This includes both the unimodal input recognizers, as for example the Automatic Speech Recognizer and the gesture recognizer, and the output generators, as the Speech Synthesizer. Moreover, it integrates the recognized inputs, assigning them temporal, logical and spatial attributes, as required by the multimodal grammar specification, and applies the production rules stored in the Multimodal Grammar Repository, to parse the multimodal input. The platform acquires the set of production rules of the grammar through the *Grammar Acquisition* component that follows an approach “by example” to allow the user to specify the multimodal sentences that have to be recognized by the system. The analysis level contains also the *Modelling components*, that are designed following a meta-model-based approach, in order to have an abstract view of the modelling process and components, by assuring, at the same time, a reasonable level of independence from the used modelling technologies. Examples of modelling components that have to be integrated in the framework are the user, content and context modelling components. Moreover, the framework should offer the possibility to extend and integrate other kinds of modelling components. Finally, the analysis level includes the *Multimodal Output Manager* for the generation (multimodal fission) of appropriate output information, through the available output modalities.
- *The planning level:* The main tasks of this level are the understanding of which is the better way to react to the user command (either directly intervening on the electro-mechanical systems, through the electro-mechanical systems Gateway, or providing specific audio/visual feedback) and the consequent adaptation of the human-machine interaction, taking into account also the outputs of the *Modelling Components*. The planning level contains the *System Response Generator*, which identifies the meta-models that are relevant for the user request and takes suitable information from these models. The *electro-mechanical systems Gateway* provides the link between electro-mechanical systems and the planning level. Proper solutions shall be applied to ensure safe interfacing and communication between the two levels.
- *The activation level:* This level contains the electro-mechanical components offering specific functionalities to the user. This level should be as independent as possible and should, in principle, work independently of the other levels. It includes a framework interface adapter offering specific functions such as communicating to the framework through the electro-mechanical systems gateway.

The core of this architecture is composed of the analysis and planning levels, which contain the fundamental elements of the framework we intend to focus our attention in this paper. In the design of the aforementioned architecture we have used an application-independent philosophy. This means that changing the I/O devices and sensors in the acquisition level and the electro-mechanical systems in the activation level allows to exploit the integrative framework in different applicative scenarios. Therefore, if we consider as application scenario a system able to recognize dangerous situations for people with neurodegenerative diseases, the architecture of Figure 2 can be instantiated as in Figure 3.

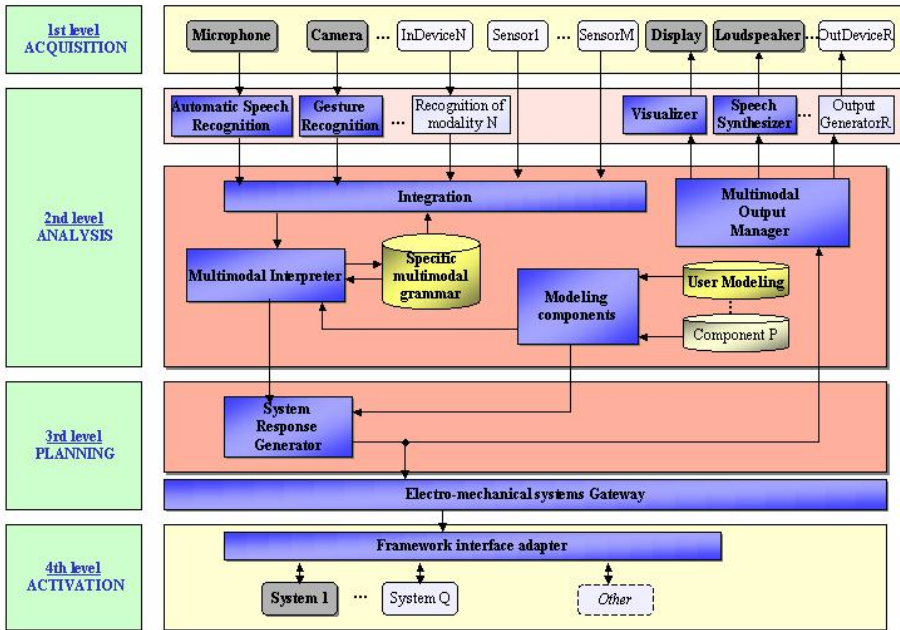


Fig. 3. Application scenario for the platform

Regarding the acquisition level, the personal assistance system makes use of two interaction modalities, which are voice and gesture. Each signal, conveyed through these modalities, is acquired by two input devices: microphone and camera, respectively. In the same way, the output message to be returned to the elderly patient is delivered through two output devices, that are the display (for visual feedback) and the loudspeakers (for synthetic voice).

The second level of the architecture is composed of a first recognition level, in which the Automatic Speech Recognizer, for the speech signal, and the gesture recognizer, for the gestural input, are placed. This application level makes use of two output generators: the visualizer and the speech synthesizer. Moreover, the second layer of the analysis level is the core part of the platform, composed of the standard components common for each application scenario, that are the Integration, the Multimodal Interpreter, and the Multimodal Output Manager. The multimodal grammar and the modelling module are the only components specific for the application scenario, which we are dealing with. In the personal assistance system we defined a user modelling module that provides the acquisition of information about patient’s needs, preferences and clinical history. On the base of the content of both the multimodal grammar and the user modelling repositories, the standard components of the analysis level modify their functioning (integration, interpretation and output management) by adapting consequently the interaction to the specific application scenario.

Analogously, the planning level is composed of the standard components common to each application scenario, that are the System Response Generator and the Electro-mechanical systems Gateway.

Regarding the activation level, the personal assistance system provides medical devices that may be activated according to the decision of the planning level.

4 The Analysis of the Multimodal Input

This section will describe in more detail the two main modules of the multimodal dialogue processing platform, that are the multimodal language interpreter and the modelling components, discussing the application scenario of monitoring patients with neurodegenerative diseases.

4.1 Multimodal Language Interpreter

Johnston and Bangalore [9] proposed a finite-state based approach for multimodal language processing, which is based on a finite-state device to parse inputs from multiple modes and to integrate them into a single semantic interpretation. To structure and interpret the multimodal commands the authors use a multimodal context free grammar that is subsequently approximated by a finite-state automaton. The authors do not provide a mechanism of grammar incremental learning.

Rudzicz [10] followed a unification-based approach, in which the fusion module applies a unification operation on multimodal inputs. This approach is based on a unification grammar that is a particular kind of context free grammar, whose rules are enriched by probabilistic aspects.

Sun et al. [11] proposed a multimodal language processor that treats multimodal language as a unique linguistic phenomenon, in which the inputs of the processor are symbols recognized by signal recognizers and the output is a syntactic structure and integrated semantic interpretation expressed in hybrid modal logic. To define multimodal inputs and specify their syntactic structures and semantic interpretations the authors use a Combinatory Categorical Grammar (CCG), a form of lexicalised grammar in which the application of syntactic rules is entirely conditioned on the syntactic type.

Starting from the assumption that speech is the predominant modality of human communication, we believe that each concept expressed by any modality can be translated into a natural language expression. Consequently, the natural language is a “ground layer” which all modalities refer to. This is the main reason that led us to extend Natural Language Processing (NLP) to the multimodal dialog processing. NLP is generally concerned with the attempt to recognize a large pattern or sentence by decomposing it into small subpatterns according to linguistic rules. In the same way, we conceived our multimodal dialogue system based on extended NLP techniques that allow to recognize multimodal sentences according to appropriate grammar rules.

In particular, the multimodal dialog definition system is based on a hybrid multimodal grammar that puts formal grammars and logical calculus together. To be more precise, we started from a kind of multidimensional grammar, termed constraints multiset grammars (CMGs), that are able to capture the structure of a variety of multimodal inputs, and, subsequently, we embedded the CMG definition into Linear Logic. The advantages of this hybrid approach over grammars are that it allows the reasoning about multimodal languages providing a prerequisite for being able to resolve ambiguities and for integrating parsing with semantic theories, and secondly it

allows to prove abstract properties of grammars as linear logic has a well-developed proof and model theory. Moreover, an algorithm for the incremental learning of the grammar is provided. This algorithm allows to add new information to the basic ground of the system enabling the recognition of not only very well defined sentences but also different sentences with a similar semantic meaning.

4.2 The Modelling Components

The multimodal systems usability and the interaction naturalness can be obtained following an *evolutionary* approach that is implemented by adaptive and adaptable interaction defined using the *Modelling Components*.

This evolutionary approach has to involve several modelling components such as the *user's model* and the context's model, enabling the automatic content and/or interaction evolution according to the context and user's differences, if these two models are available and maintained.

Focusing on the user's model, it contains information about the user known to the system. Therefore, the Modelling Component module acquires and maintains information of the user's model and the other models during the use of the system.

We accomplish this goal considering the multimodal interaction and communication as a process occurring on two levels:

- 1) the first level has to permit to describe the users' features and constraints (using a general framework) on them not for a specific user and not for a specific context, and
- 2) the second level, which takes into account the domain, the context and the user level processing, gives the description of the specific users' model.

The first level has to permit to acquire the second one, because it is the metamodel containing all the features involved in user modelling.

Typically the user's model has to contain information such as:

- user preferences
- user interests,
- user attitudes,
- user knowledge and
- user goals.

This information has to be produced using a general framework starting from information acquired by the interaction and communication process, so that the user's model is produced.

In the framework definition we have supposed that users' preferences are resulting by values of such parameters such as *physical states*, *emotional states*, or *personality*, and their constraints.

All these information must be put in relation with other models maintained by the system.

5 Conclusions and Future Work

In this paper we have presented a novel general framework for multimodal dialogue processing, which is conceived following an application-independent philosophy. In fact, it is able to manage multimodal communication between people and the environment in different application scenarios.

The core of this framework is composed of the multimodal language interpreter and the modeling component. The former has been developed by using a grammar-based approach, that puts formal grammars and logical calculus together. In particular, a specific kind of multidimensional grammar, constraints multiset grammars (CMGs), are used to capture the structure of a variety of multimodal inputs, and, subsequently, the CMG definition is embedded into Linear Logic. The advantage of this approach is that it allows the reasoning about multimodal languages providing a prerequisite for being able to resolve ambiguities and for integrating parsing with semantic theories. The modeling component is designed following a meta-model-based approach, in order to have an abstract view of the modelling process and components, and assuring, at the same time, a reasonable level of independence from the used modelling technologies.

We applied this theoretical platform for the definition and interpretation of the dialogue between a human and a particular kind of personal assistance system that is able to monitor and recognize dangerous situations for people with neurodegenerative diseases.

As future work we would perform an evaluation process in order to calculate the effectiveness of our approach and to compare it to other existing approaches.

References

1. Coutaz, J., Caelen, J.: A Taxonomy For Multimedia and Multimodal User Interfaces. In: Proceedings of the 1st ERCIM Workshop on Multimedia HCI, November 1991, Lisbon (1991)
2. Schomaker, L., Nijtmans, J., Camurri, A., Lavagetto, F., Morasso, P., Benoit, C., Guiard-Marigny, T., Le Goff, B., Robert-Ribes, J., Adjoudani, A., Defee, I., Munch, S., Hartung, K., Blauert, J.: A Taxonomy of Multimodal Interaction in the Human Information Processing System. In: Multimodal Integration for Advanced Multimedia Interfaces (MIAMI). ESPRIT III, Basic Research Project 8579 (1995)
3. Bolt, R.: Put-that-there: Voice and gesture at the graphics interface. *Computer Graphics* 14(3), 262–270 (1980)
4. Neal, J.G., Shapiro, S.C.: Intelligent multimedia interface technology. In: Sullivan, J., Tyler, S. (eds.) *Intelligent User Interfaces*, pp. 11–43. ACM Press, New York (1991)
5. Nigay, L., Coutaz, J.: A generic platform for addressing the multimodal challenge. In: *The Proceedings of the Conference on Human Factors in Computing Systems*, ACM Press, New York (1995)
6. Cohen, P.R., Johnston, M., McGee, D., Oviatt, S.L., Pittman, J., Smith, I.A., Chen, L., Clow, J.: Quickset: Multimodal interaction for distributed applications. *ACM Multimedia*, 31–40 (1997)
7. Vo, M.T.: A framework and Toolkit for the Construction of Multimodal Learning Interfaces, PhD. Thesis, Carnegie Mellon University, Pittsburgh, USA (1998)

8. Wahlster, W., Reithinger, N., Blocher, A.: SmartKom: Multimodal Communication with a Life-Like Character. In: Proceedings of Eurospeech, Aalborg, Denmark (2001)
9. Johnston, M., Bangalore, S.: Finite-state Multimodal Integration and Understanding. *Journal of Natural Language Engineering* 11(2), 159–187 (2005)
10. Rudzicz, F.: Clavius: Bi-directional Parsing for Generic Multimodal Interaction. In: Proceedings of COLING/ACL 2006, Sydney (2006)
11. Sun, Y., Shi, Y., Chen, F., Chung, V.: An efficient unification-based multimodal language processor in multimodal input fusion. In: Proceedings of the 2007 Conference of the Computer-Human interaction Special interest Group (Chisig) of Australia on Computer-Human interaction: Design: Activities, Artifacts and Environments, OZCHI 2007, Adelaide, Australia, November 28-30, 2007, vol. 251, pp. 215–218. ACM, New York (2007)

A Comparison of Microphone and Speech Recognition Engine Efficacy for Mobile Data Entry

Joanna Lumsden, Scott Durling, and Irina Kondratova

National Research Council of Canada, IIT e-Business, 46 Dineen Drive, Fredericton,
N.B., Canada E3B 9W4
{jo.lumsden, scott.durling, irina.kondratova}@nrc-cnrc.gc.ca

Abstract. The research presented in this paper is part of an ongoing investigation into how best to incorporate speech-based input within mobile data collection applications. In our previous work [1], we evaluated the ability of a single speech recognition engine to support accurate, *mobile*, speech-based data input. Here, we build on our previous research to compare the achievable speaker-*independent* accuracy rates of a variety of speech recognition engines; we also consider the relative effectiveness of different speech recognition engine and microphone pairings in terms of their ability to support accurate text entry under realistic mobile conditions of use. Our intent is to provide some initial empirical data derived from mobile, user-based evaluations to support technological decisions faced by developers of mobile applications that would benefit from, or require, speech-based data entry facilities.

Keywords: mobile speech input, microphone efficacy, speech recognition accuracy/efficacy, mobile technology, mobile evaluation.

1 Introduction

Although speech recognition has been nominated as a key potential interaction technique for use with mobile technologies [2-4], its widespread commercialization and adoption remains limited on account of unacceptable error rates [5]. It is estimated that accuracy rates can drop by as much as 20%-50% when speech is used in natural environments [3-5]. Achievable accuracy is a strong determinant of users' perception of speech recognition acceptability [2]; as such, it is important that we address the challenge of developing *effective* speech-based solutions for use in mobile settings.

A number of measures can be taken to increase recognition accuracy [2, 5-9]. The research presented in this paper focuses on two such measures: (1) empirically-based, context-specific selection of speech recognition engines (and microphones) to maximize potential accuracy within a given domain of use; and (2) identification of the effect of background noise and mobility on speech recognition accuracy. Specifically, we report on a comparison of the capacity of 5 different speech recognition engines to support accurate, mobile, speech-based text entry. The work discussed in this paper represents a continuation of our ongoing investigation in this area [1]. Our previous work reported on an evaluation of the ability of a single speech recognition engine (SRE) to support accurate, mobile, speech-based data input; here, we build on

our previous research to compare the achievable accuracy rates of a variety of SREs. In the following sections, we briefly describe the background to our work (Section 2) and the evaluation design and process (Section 3); we would refer interested readers to [1] for greater detail in both regards. In Section 4 we present, in detail, our results. In Section 5 we draw some *initial* conclusions from our observations.

2 Related Work

Speech recognition accuracy is typically degraded in noisy, mobile contexts because not only does background noise contaminate the speech signal received by the SRE but, additionally, people modify their speech under noisy conditions [5]. Specifically, in noisy environments, speakers exhibit a reflexive response known as the Lombard Effect [5, 10, 11] which causes them to modify the volume at which they speak and to hyperarticulate words. Since research suggests it is not possible to eliminate or selectively suppress Lombard speech, the onus is placed on SREs to be able to cope with variations in speech signals caused by mobile speech input under noisy and changing acoustic conditions [5, 10].

Under mobile conditions, background noise can confuse, contaminate, or even drown out a speech signal; as a result, SRE accuracy has been shown to steeply decline in even moderate noise [4, 5]. Even in stationary conditions, microphone type, placement, and quality affects user performance [12]; when the complexities of non-static usage environments are introduced, the influence of the microphone becomes even more pronounced [1, 5, 7-9]. Our previous research focused on assessing the impact of mobility and background noise on the efficacy of three different microphones [1] for supporting mobile speech-based data entry. Although previous research (see [1] for a detailed review) had been conducted into (a) the impact of mobility and background noise on speech recognition, and (b) the influence of microphone type on speech recognition, our prior work brought together, into one *novel* evaluation, many of the constituent – and previously unconnected – elements of previous studies to empirically compare the ability of three different microphones (appropriate in terms of form and function) to support accurate speech-based input under *realistic, mobile, noisy conditions*. In the research we present here, we extend the reach of our previous study to apply the input signals we recorded in our initial study to a series of SREs in order to compare their efficacy to support accurate speech-based input under realistic, mobile, noisy conditions.

3 Evaluation Design and Process

There are two components to describing our evaluation design and process: (a) the set-up from our previous study; and (b) the manner in which we extended the previous study to complete the research we report on here. In the following sections, we provide a brief overview of the former (for extensive detail, see [1]) and then describe how we used the data collected in (a) to extend our analysis to compare multiple SREs.

3.1 Previous Experimental Design

Our previous study compared three commercially available and commonly used microphones – the NextLink Invisio Mobile (bone conduction) microphone [13], Shure’s QuietSpot QSHI3 [14], and the Plantronics DSP-500 microphone [15] – in terms of their efficacy to facilitate mobile speech input. We developed a very simple data input application which ran on a tablet PC running Windows XP and used IBM’s ViaVoice [16] speaker-independent SRE with a *push-to-talk* strategy. For each data entry item, participants were shown precisely what to enter, and given a maximum of three attempts in which to achieve an accurate entry. Participants were given training on how to use the system (in conditions commensurate with the actual experimental set-up) prior to commencing the study tasks.

We adopted a counterbalanced, between-groups design whereby participants were allocated to groups partitioned according to the three microphones; in counterbalanced order, each participant was required to complete a series of 10 data entry items under quiet environmental conditions, and 10 data entry items when surrounded by recorded city street sounds played at 70dB using a 7.1 surround sound system in our lab. While completing their data entry tasks, participants were required to be mobile using our ‘hazard avoidance’ or ‘dynamic path system’ – see [1] for more details.

Twenty four people participated in our study, 8 per microphone group. Since studies have shown that speech recognition rates are typically much lower for accented speakers [5], we restricted our recruitment to participants who were native English speakers with a Canadian accent; we recruited equal numbers of male and female participants, but restricted our age range to 18 – 35 year olds (young adults) to limit speaker variation that comes with age. In placing these restrictions on our participant recruitment, we recognize that we limited our ability to generalize from our results but we wanted to reduce the extraneous factors that *may* have impacted our results such that we were able to focus on the effects of the *microphones* rather than variances between the speakers; additionally, SREs available in North America are typically optimized to a ‘generic’ North American accent so by placing these restrictions on the participant recruitment we effectively tested the speech recognition technology within its self-proclaimed bounds.

Of the range of measures we recorded for analysis during our previous study, the data of interest to the study reported here is the series of speech signal (or voice) recordings upon which our SRE operated. In essence, each of these recordings captures what participants said, together with the background noise picked up by their respective microphones. In so doing, these recordings capture the presence of Lombard speech as well as the impact being mobile (i.e., walking) had on participants’ speech: that is, the recordings are representative of likely speech patterns in real world contexts where users are exposed to noisy conditions as well as required to be mobile while interacting with their mobile device. The following section describes how we utilized these recordings in our current study.

3.2 Current Study Design

As already mentioned, the intent of our current study was to extend our analysis to compare the efficacy of a range of SREs to that of the one used in our previous study

(i.e., IBM's ViaVoice), as well as to determine if there was a single microphone-SRE pairing that proved to be most effective. Bearing in mind that we set up our previous experiment to reflect the real world context in which speech recognition may ultimately be used, and given that we took care to homogenize our participant group as far as possible with respect to accent and age, we feel that the results of our current investigation are valid as an *initial indication* of the potential benefits of one SRE over another for mobile interaction; that said, we suggest that the results of this study be considered as a baseline and acknowledge that the impact of speaker accent, especially, and age need to be investigated independently.

We compared the *speaker-independent* efficacy of (1) the IBM ViaVoice SRE (from our previous study) to the efficacy of 4 mainstream SREs: (2) the speaker-independent Sphinx 4 open source SRE (developed by CMU), set up with the default configuration and loaded with the specific grammars (as before) that were needed for the data entry application [17]; (3) Philips' SRE [18] and (4) Microsoft's SRE within Windows XP [19], both of which were set up with our required grammars; and (5) the Nuance Dragon Naturally Speaking SRE [20], which was essentially grammarless because it did not support grammar specification. Engines 3, 4, and 5 all support *speaker-dependence* (i.e., can be trained for specific users) but we used each in a *speaker-independent* mode in order to assess the 'walk-up-and-use' capabilities of each; we loaded a fresh profile for each participant such that the engines did not learn over time across participants.

We focused on first-attempt data input: that is, we filtered out, and only used, the recordings associated with participants' first attempts (successful or not) at inputting each data item. This decision not only allowed us to accommodate the fact that, for items correctly recognized first time by ViaVoice in our previous study, we only had one recording, but it also placed all our SREs on an equal footing.

We passed, using an automated process, the first-attempt voice recordings through each of our 4 additional SREs to derive a Boolean measure of accuracy for each data entry attempt. Our set of recordings included 160 files per microphone, 80 recorded under our quiet condition and 80 under our noisy condition. On this basis, each SRE was subjected to a total of 480 recordings, allowing us to test each against the three microphones and the two background audio conditions.

4 Results and Discussion

Our primary measure of accuracy was calculated as a ratio of the total number of first-attempt correct entries divided by the total number of tests (according to analytical breakdown). A multiple factor ANOVA showed that SRE ($F_{4,2370}=27.03$, $p<0.001$) and the combination of SRE and microphone ($F_{8,2370}=3.49$, $p=0.001$) had a significant effect on accuracy. Figure 1 shows the accuracy rate achieved according to SRE. Tukey HSD tests showed that: the accuracy achieved using IBM ViaVoice was significantly higher than for all of the other engines; that the speech recognition accuracy achieved using Philips' SRE was significantly less than all of the other engines; and that the difference in accuracy achieved using the Sphinx, Microsoft, and Dragon engines was not statistically significant.

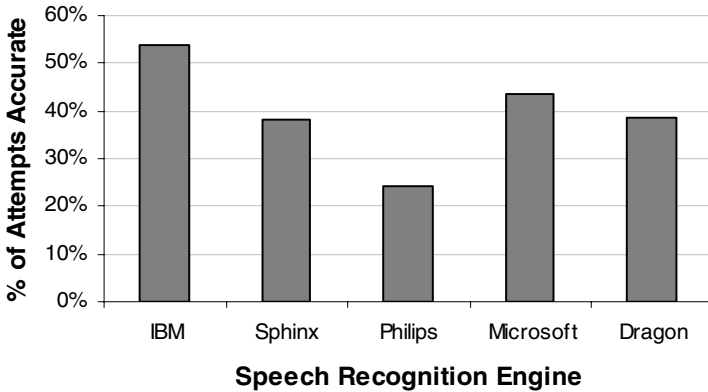


Fig. 1. Accuracy rate (accurate attempts/total attempts) according to speech recognition engine

As can be seen from Figure 1, the maximum accuracy rate achieved on first attempt was approximately 54% for the IBM ViaVoice engine.

The results shown in Figure 1 are calculated irrespective of background noise. We did not find the combination of SRE and noise level to significantly affect accuracy ($F_{4,2370}=0.42$, $p=0.796$); furthermore, by analyzing the data across all background noise conditions, we obtain a picture of the likely average accuracy achievable by mobile or nomadic users who may typically move between noisy and quiet environments as they work.

Figure 2 shows the accuracy rates achieved according to SRE+microphone pairing. Tukey HSD tests showed that, with the exception of the Philips' SRE, when combined with the Invisio microphone, each SRE returned significantly lower accuracy rates than when combined with the other two microphones; there was no significant difference for these 4 SREs when combined with the QSHI3 and DSP-500 microphones. In the case of the Philips' SRE, however, the Invisio+SRE combination only returned significantly lower accuracy rates than the DSP-500+SRE combination ($p<0.001$); the Philips' SRE+QSHI3 and SRE+DSP-500 combinations did, however, return significantly different accuracy rates ($p<0.001$).

Focusing on the Invisio microphone across all 5 SREs, the Invisio+IBM and Invisio+Philips combinations returned significantly different accuracy rates ($p<0.001$), the former demonstrating a higher accuracy rate; the same was true for the Invisio+IBM and Invisio+Dragon combinations ($p=0.01$), for the Invisio+Microsoft and Invisio+Philips combinations ($p<0.001$), and for the Invisio+Microsoft and Invisio+Dragon combinations ($p=0.01$). When combined with the Invisio microphone, there was no significant difference in the accuracy rates returned by the IBM, Sphinx, and Microsoft SREs.

The QSHI3+Philips combination was significantly less accurate than all of the other SREs combined with the same microphone ($p<0.001$ in each case); additionally, the QSHI3+IBM combination was significantly more accurate than the QSHI3+Sphinx combination ($p=0.016$). With these noted exceptions, there were no other significant differences for the QSHI3 microphone across the various SREs.

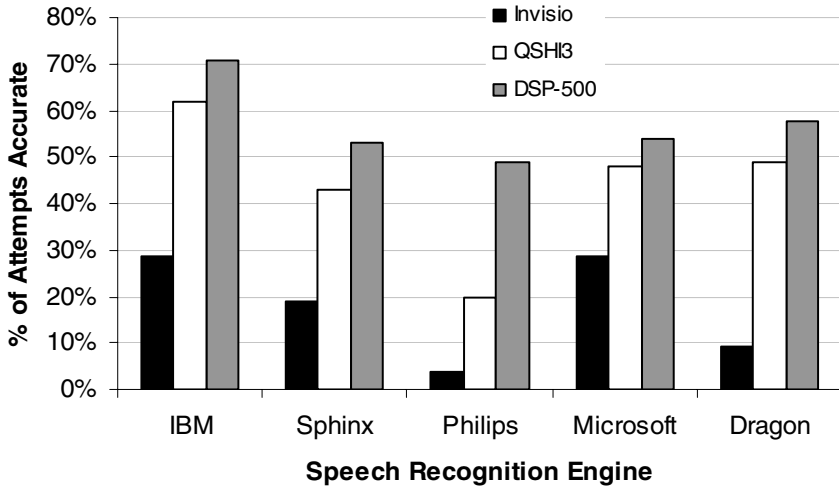


Fig. 2. Accuracy rate according to speech recognition engine and microphone

When combined with the IBM SRE, the DSP-500 microphone was significantly more accurate than when combined with the Sphinx ($p=0.04$) and Philips ($p<0.001$) SREs. There were no other significant comparisons across any of the other pairings for this particular microphone – most noticeably, unlike its performance with the other two microphones, the Philips SRE was on a par with the majority of the other SREs when combined with the DSP-500 microphone.

With the exception of the QSHI3+IBM, DSP-500+Microsoft, and DSP-500+Dragon combinations, the combination of DSP-500 microphone and IBM ViaVoice SRE returns the highest overall accuracy rates (approximately 71%). These results demonstrate the impact of pairing microphones with SREs to achieve the best possible potential for accurate speech recognition: for example, where ViaVoice’s dominance is significantly reduced when paired with the Invisio microphone, the Philips’ generally poor recognition is greatly boosted when paired with the DSP-500 microphone.

Of the total 952 accurately recognized data inputs, 41 were correctly recognized *despite* user error during input. We classify user error as instances where: users pressed the push-to-talk button but didn’t speak (this was registered as an input attempt by the system); users pressed the push-to-talk button after they had started to speak or released it before they finished speaking (essentially clipping their recorded speech); or users simply said the wrong thing. Figure 3 shows the extent to which each microphone+SRE coped with such errors to return a correct interpretation of the users’ input.

Although we attribute no statistical significance to the tallies shown in Figure 3, it is interesting to note that certain microphone+SRE combinations appear better able to accommodate user input error. In particular, the fact that 12% of the correctly recognized flawed inputs are attributable to the Philips SRE+DSP-500 combination reflects, and perhaps accounts for, the significantly better accuracy rate achieved by this

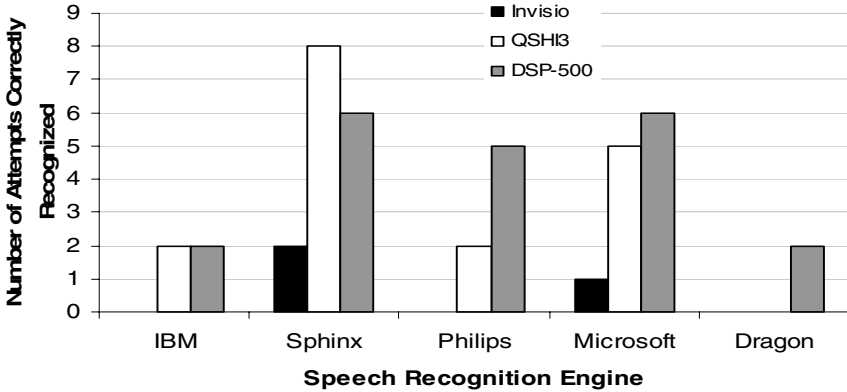


Fig. 3. Breakdown of correct recognition despite user input error

combination compared to the SRE's pairing with the other microphones (see Figure 2). Furthermore, although the IBM ViaVoice+DSP-500 combination has thus far excelled compared to the other SRE+microphone combinations, its dominance is much less when required to cope with flawed user inputs: in fact, the Sphinx+QSHI3 dominates in this capacity.

5 Conclusions and Preliminary Guidelines

Although it is premature to suggest concrete guidelines on the basis of our initial research, we close this paper with our conclusions and some preliminary guidance (which we call suggestions to reflect their current status) for designers.

Suggestion 1: Carefully consider the selection of microphone+SRE pairing, preferably by conducting empirical comparisons relative to intended context of use. Our results clearly indicate the importance of carefully considering SRE+microphone pairings relative to a specific context of use when developing speech-based mobile applications. By simply changing the microphone with which a given SRE is paired, it is possible to dramatically enhance the achievable accuracy rates – for example, ViaVoice paired with the Invisio microphone returned a deplorable accuracy rate of 29% but this more than doubled to 71% when the same SRE was paired with the DSP-500 microphone. We have only compared microphone+SRE pairings relative to one context of use so are not in a position to make generalized recommendations concerning microphone+SRE pairings relative to various contexts of use at this time; furthermore, we would strongly encourage designers to conduct contextually-relevant, empirical analysis relative to the *specific* context for which they are designing a mobile application in order to elicit the most reliable data and thereby make the most informed decision.

Suggestion 2: Determine the likely extent to which target users will make mechanical or verbal errors during input (i.e., to what extent their physical environment and/or multitasking behavior may impact their capacity to devote attentional

resources to speech-based input) and be prepared to trade off general accuracy against error tolerance. We have demonstrated the importance of considering, for any given application and domain, the extent to which users are likely to make mechanical or verbal errors during speech-based data entry. Our results suggest that designers may, depending on the context for which they are designing, have to make trade offs between SRE-microphone pairings that return high raw accuracy and pairings that have an increased ability to cope with flawed input.

Suggestion 3: Carefully consider the requirement for accurate first time data entry versus scope to tolerate repeated entries in order to enter data correctly. Our study only looked at first attempt accuracy; while this is often essential, we recognize that under situations where multiple attempts to achieve an accurate input would be tolerable, the breakdown of ultimate accuracy rates across the SREs we tested might differ.

Suggestion 4: Carefully consider the applicability of different microphone designs relative to the intended context of use. We recommend that designers carefully consider not only the accuracy that can be achieved using a given microphone, but also its appropriateness to the context in which it is to be used – e.g., if a user has to wear a safety helmet or to use specific equipment such as a stethoscope, to what extent can a given headphone mounted microphone be accommodated or does an alternative form factor need to be sought? Accuracy alone is insufficient to make the microphone usable.

Finally, as previously discussed, our study is not without its limitations; as such, we present our results within the scope of our noted caveats. We were testing speaker-independent operation of the SREs (since our interest was in their ‘walk-up-and-usability’); we recognize that if the speaker-dependent SREs (Philips, Microsoft, and Dragon) were to be trained, their accuracy rates would likely dramatically improve. That being said, our results not only demonstrate the difference in the capabilities of these systems (which are normally trained prior to use) to cope with speaker-independent, walk-up-and-use situations, but we also present the results as empirical data to assist a designer when selecting an SRE and microphone for use in a speaker-independent capacity. At the very least, we have empirically highlighted the complexity of decisions surrounding microphones and SREs for mobile applications; we have provided data that was not previously available to designers and, as such, hope that it not only proves useful to designers of speech-based mobile data input, but also highlights those areas that require detailed consideration when making speech technology decisions during the design process.

References

1. Lumsden, J., Kondratova, I., Durling, S.: Investigating Microphone Efficacy for Facilitation of Mobile Speech-Based Data Entry. In: Proceedings of British HCI 2007 Conference, Lancaster, UK, September 3-7, pp. 89–98 (2007)
2. Price, K., Lin, M., Feng, J., Goldman, R., Sears, A., Jacko, J.: Data Entry on the Move: An Examination of Nomadic Speech-Based Text Entry. In: Stry, C., Stephanidis, C. (eds.) UI4ALL 2004, vol. 3196, pp. 460–471. Springer, Heidelberg (2004)

3. Sawhney, N., Schmandt, C.: Nomadic Radio: Speech and Audio Interaction for Contextual Messaging in Nomadic Environments. *ACM Transactions on Computer-Human Interaction* 7(3), 353–383 (2000)
4. Ward, K., Novick, D.: Hands-Free Documentation. In: *Proceedings of 21st Annual International Conference on Documentation (SIGDoc 2003)*, San Francisco, USA, October 12–15, pp. 147–154 (2003)
5. Oviatt, S.: Taming Recognition Errors with a Multimodal Interface. *Communications of the ACM* 43(9), 45–51 (2000)
6. Lumsden, J., Kondratova, I., Langton, N.: Bringing A Construction Site Into The Lab: A Context-Relevant Lab-Based Evaluation Of A Multimodal Mobile Application. In: *Proceedings of 1st International Workshop on Multimodal and Pervasive Services (MAPS 2006)*, Lyon, France, June 29, pp. 62–68 (2006)
7. Sammon, M., Brotman, L., Peebles, E., Seligmann, D.: MACCS: Enabling Communications for Mobile Workers within Healthcare Environments. In: *Proceedings of 8th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 2006)*, Helsinki, Finland, September 12 - 15, pp. 41–44 (2006)
8. Sebastian, D.: Development of a Field-Deployable Voice-Controlled Ultrasound Scanner System, M.Sc. Thesis, Dept. of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA, USA (2004)
9. Vinciguerra, B.: A Comparison of Commercial Speech Recognition Components for Use with the Project54 System, M.Sc. Thesis, Dept. of Electrical Engineering, University of New Hampshire, Durham, NH, USA (2002)
10. Pick, H., Siegel, G., Fox, P., Garber, S., Kearney, J.: Inhibiting the Lombard Effect. *Journal of the Acoustical Society of America* 85(2), 894–900 (1989)
11. Rollins, A.: Speech Recognition and Manner of Speaking in Noise and in Quiet. In: *Proceedings of Conference on Human Factors in Computing Systems (CHI 1985)*, San Francisco, USA, April 14 - 18, pp. 197–199 (1985)
12. Chang, J.: Speech Recognition System Robustness to Microphone Variations, M.Sc. Thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA (1995)
13. NextLink, Invisio Pro, <http://www.nextlink.se/>
14. Shure, QuietSpot QSHI3, <http://www.sfm.ca/quietspot/qshi3.html>
15. Plantronics, DSP-500 Headset, http://www.plantronics.com/north_america/en_US/products/cat640035/cat1430032/prod440044
16. IBM, Embedded ViaVoice, http://www-306.ibm.com/software/pervasive/embedded_viavoice/
17. CMU, Sphinx-4, <http://cmusphinx.sourceforge.net/sphinx4/>
18. Philips, Speech SDK, <http://www.speechrecognition.philips.com/index.asp?id=521>
19. Microsoft, Windows Desktop Speech Technology, <http://msdn.microsoft.com/en-us/library/system.speech.recognition.aspx>
20. Nuance, Dragon Naturally Speaking, <http://www.nuance.com/naturallyspeaking/sdk/client/>

A GRID Approach to Providing Multimodal Context-Sensitive Social Service to Mobile Users

Massimo Magaldi, Roberto Russo, Luca Bevilacqua, Stefania Pierno, Vladimiro Scotto di Carlo, Fabio Corvino, Luigi Romano, Luisa Capuano, and Ivano De Furio

ENGINEERING.IT S.p.A. Italia

{massimo.magaldi, roberto.russo, luca.bevilacqua, stefania.pierno, vladimiro.scottodicarlo, fabio.corvino, luigi.romano, luisa.capuano, ivano.defurio}@eng.it

Abstract. In this paper, we describe a grid approach to providing multimodal context-sensitive social services to mobile users. Interaction design is a major issue for mobile information system not only in terms of input-output channels and information presentation, but also in terms of context-awareness. The proposed platform supports the development of multi-channel, multi-modal, mobile context aware applications, and it is described using an example in an emergency management scenario. The platform allows the deployment of services featuring a multimodal (synergic) UI and backed up on the server side by a distributed architecture based on a GRID approach to better afford the computing load generated by input channels processing. Since a computational GRID provides access to “resources” (typically computing related ones) we began to apply the same paradigm to the modelling and sharing of other resources as well. This concept is described using a scenario about emergencies and crisis management.

1 Introduction

The penetration of mobile device in western countries is high and still increasing. At the same time new generation terminals feature ever increasing computing power, opening new possibilities for innovation, especially in service delivery.

One emerging trend about service evolution is for services to cater not only to individuals but also to communities of users. Communities are a social phenomenon where people with common interests, experiences, and objectives are brought together. They provide a social place where individuals exchange and share information, knowledge, and emotions and jointly undertake activities. Managing the creation or deletion of flexible (possibly ad-hoc) communities improves the user experiences in communities [1].

MoSoSo (Mobile Social Software), is a class of mobile applications that aims to support social interaction among interconnected mobile users [2]. While existing Internet-based services have already shown the growing interest in communication support for communities, *MoSoSo* adds additional dimensions to group communication by exploiting contextual data such as time and geographical location.

When designing *MoSoSo* applications, three important differences between desktop and mobile environments should be taken into account:

- The physical context of use is no longer static and implies some constraint to user attention;
- The social context also becomes dynamic: mobile communities member are tied up by common interest and contextual information, like location and time;
- MoSoSo applications are designed not just for communication but for usage in everyday life situations: users are always socially connected.

In our vision, incorporating MoSoSo in public services area could lead to extremely innovative mobile services, leveraging on dynamic management of *ad-hoc* communities, context-awareness (i.e. time and location), user profile management and multimodal interaction.

One domain where such benefits will matter most, could be emergencies and crisis management. In fact, emergency response operations typically implies the coordination of physical resources, personnel and volunteers belonging to different organizations in contexts where ineffective operations can cause loss of lives.

From an IT standpoint, this requires services and resources sharing across typically heterogeneous hardware and software environments belonging to different organizations. The Virtual Organizations paradigm address this issue: “VOs enable disparate groups of organizations and/or individuals to share resources in a controlled fashion, so that members may collaborate to achieve a shared goal” [4]. In those circumstances dynamism, flexibility and interoperability become essential requirements.

Interoperability, in particular, is a key issue in e-Government domain due to the increasing demand for integrated services. We aims to integrate multimodal mobile social application‘ users into typical Grid resource management model. To this end, we have designed an experimental platform that support the development of multimodal MoSoSo application, allowing for an easy integration of mobile community users into Grid based VO.

OGSA (Open Grid Service Architecture [6]), a refinement of the SOA concept, allows interoperability of “resources”. In fact the OGSA specification allows each resource to be seen as a service with a standard interface. In the WS-Resource framework conceptual model, a Web service is a stateless entity that acts upon, provides access to, or manipulates a set of logical stateful resources (documents) based on messages it sends and receives [7][8].

Obviously while evolving from SOA to OGSA all architectural components must be extended to deal not only with services but also with resources. For example, as far as processes are concerned, the workflow engine has to be able to compose both. Similarly a logical enhancement to the UDDI registry is required to store information about WS-Resources too.

Whereas there are interesting technology products (both commercial and open sources) dealing with the multimodal client interfaces [16][17][18][19] and grid middleware [4][6][20][21], the innovative idea proposed in this article is bringing them together in order to enable innovative social services like multimodal emergency services and reduce digital divide.

The rest of this paper is organized as follows. Next section describes multimodal part of the overall architecture (front-end). Then it will be described the back-end architectural solution based on grid paradigm. Final remarks in the last section conclude the paper.

2 Multimodal Architecture Overview

The multimodal part of the overall architecture (figure 1, figure3), is composed by the following modules:

- Front end: collects input from clients and routes it to recognizers (speech, sketch and handwriting recognizers);
- Fusion: semantically merges recognized input fragments coming from different channels;
- Business Logic: selects appropriate contents;
- Fission: sends the selected content to final users.

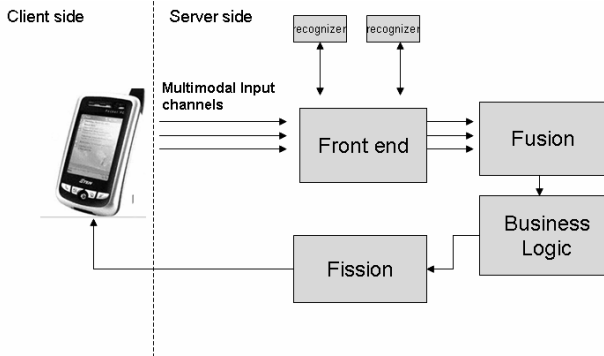


Fig. 1. Architecture overview

Although the fundamental architectural idea is known [22], we enriched it with web 2.0, telecommunication & grid technologies, open to community model based interaction. We think the final results is interesting.

In the next sections we will expand on this evaluation. We will first describe how we managed to assemble on-the fly mobile multimodal user interfaces using thin clients that exploit resources available on the network. Then, we will present the back-end architecture that deals with the induced computational load .

We focused on using commonly available mobile devices (PDAs or smart phones), and tried to avoid installing specific sw environments. Hence we had to develop a very light software framework for building the multimodal interface. The framework is based on several standard protocols, over available networks, which allow to interact with both local environment and remote servers.

Being multimodal, such an interface had to be able to:

- collect multimodal input from different channels: speech, sketch, handwriting;
- render outputs selecting the best possible output channel (multimedia output).
- exploit local resources to create an appropriate context for service fruition.

Since we chose to use light, standard “thin” terminals, all collected input fragments had to be routed to network based modal recognizers. All those functions are grouped in a small footprint application: MMUIManager (MultiModal User Interface

Manager). The MMUIManager, like a browser, receives from servers information describing the UI look and feel, loading locally just the minimum software layer needed to support the desired user interaction. Similarities stop here, though: a thin client based MultiModal User Interface Manager is technologically far more complex than a browser.

Realizing it was a challenging task. Suffice it to say that, since a standard markup language for synergic mobile multimodal UI has not emerged yet, we had to develop our own: an XML based markup language for aggregating multimodal objects: LIDIM (from the Italian Linguaggio di Definizione Interfaccie Multimodali, - language for designing multimodal interfaces).

The basic idea was to develop a framework and a related tag language able to deal with any potential combination of input modes and able to compose multimodal output objects.

The thin client approach implies that the MMUIManager has to be connected to recognition server. To avoid local buffering of input signals we adopted the telecommunication protocols SIP/RTP and developed special multimodal objects able to manage streaming protocols both in input and in output.

2.1 Mobile Interface: The Thin Client Approach

The main MMUIManager tasks are:

- build the graphic interface using an XML user interface language;
- manage the input and output channels on the device;

The UI needs to be able to simultaneously collect multimodal input and show multimedia output according to instructions coming from the server.

This approach has a significant advantage: the very same MMUI manager can be used in very different applications contexts content or logic change will be only necessary on server side.

It is a crucial advantage for portable terminals whose users are not, and need not to become, used to installing and configuring local sw applications.

According to our goals, the MMUIManager must be based on an engine for interpreting an XML language for creating multimodal/multimedia interface on the fly. Considering this, we have designed and developed a framework for aggregating multimodal objects. A composition of multimodal objects creates a complete multimodal user interface.

The interaction modes enabled by our multimodal framework are, at the moment, the following:

- point on specific buttons (ex. a “Back” button) and point on object;
- draw/handwrite on the screen;
- speak.

Output media supported are: image, text, Html, audio, video.

2.2 Input Channels

We want to manage synchronous coordinate synergic multimodality, thus the client has to collect many different input channels at the same time.

At the actual state of development the inputs channels that can be combined are:

- Speech
- Pointing
- Sketch
- Handwriting

The acquisition of “pointing” is the simplest. Every time the user touches a sensible on screen object an HTTP call containing the object identifier is sent to the server to be merged with other concurrent modal fragments.

The acquisition of sketch/handwriting is more complex. The client side framework defines special multimodal objects sensible to stylus inputs and traces. Those traces are buffered locally and sent to the front-end server via HTTP using a standard format (InkML). Front-end server, in turn, is configured for routing the input acquired to a recognizer and to the fusion module.

Automatic Speech Recognition is the most complex

Opening a speech stream an user can send speech commands for server side recognition. To trigger the recognition process start we introduced a special “start” keyword. The recognition ends after a pause in user speech

2.3 Output Channels

The output channels that the MMUIManager can process are depicted in Fig. 2.

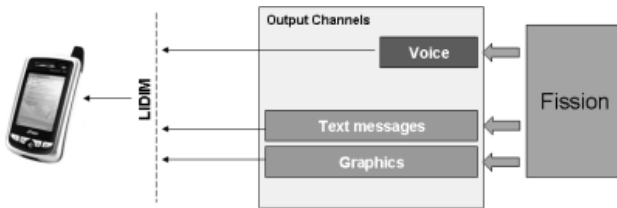


Fig. 2. Output channels schema

The voice output channels can play either pre-recorded audio streams or text streams converted in voice (TTS) by a specific engine.

According to our light-weight approach, the LIDIM file just contains information about resource location (an address). The MMUIManager takes care of retrieving them using HTTP or RTP.

The text Message channel allows showing popup text messages that can also be used for asking the user for confirmations.

The Graphics channel instructs the MMUIManager about the graphical interface layout to be shown. All the output information are included in a LIDIM page and the Fission module takes care of sending it to the MMUIManager.

It is worth noticing that the fission module is completely asynchronous. This is an important feature: real adaptive services must be able to react by adapting the user interface to context changes, even without an explicit user request, typically to react to usage context changes (noise or lightning conditions).

3 Back-End Architectural Overview

Fig. 3 shows the logical view of our back-end architecture.

The applications, being highly service specific, will not be discussed.

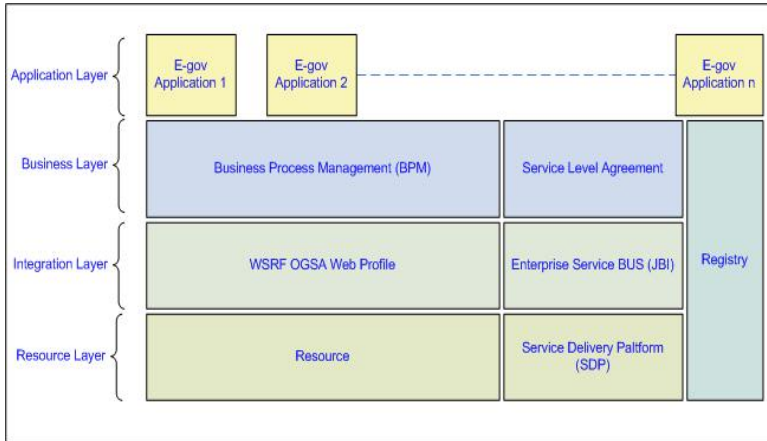


Fig. 3. Back.End (SIEGE) architecture

The Business Process Management (BPM) takes care of e-Government processes, dealing with process flow design and execution. To enable processes combining both WS and WS-Resources, we selected WS-BPEL [10] as script language and we are hence investigating how to extend it to fully support WSRF-compliant services [11].

Processes that can be fully defined at design time do not pose significant research challenges, hence we concentrated our research efforts on processes that need to be planned dynamically at execution time. To this aim we are investigating different artificial intelligence techniques that will leverage semantic descriptions of both services and resources. This automatic process planning would either adapt an existing template (stored in a repository) or try to compose it ex novo.

A Service Flow Planner component in the BPM is devoted to this task while a Match-Maker component cooperates by finding the best service/resource available (early binding). These component are fundamental in crisis management scenarios, where processes need to be highly dynamic.

While the Business Management System takes care of the high level coordination of services it does not need to concern itself with lower level details.

Functions such as data transformation, intelligent routing based on message content, protocol translation, message AAA (authentication, authorization, accounting) management, transaction management are best taken care of by a specialized component: the Enterprise Service Bus (JBI).

In distributed heterogeneous systems, mechanisms and technologies able to support a fast and effective services/resources discovery (an effective discovery mechanism will have to deal with services/resources capabilities and resources status) are fundamental. In literature the use of semantic description is widely considered as the most

promising approach about this. In particular the Semantic-OGSA [9], proposal treats metadata as a first class citizen while defining a set of services suited for metadata management (lifetime management, change notification, etc).

Semantic-OGSA is particularly interesting in that it provides for a flexible introduction of semantic data in the architecture: grid resources extended with semantic descriptions may operate together with grid resources that do not receive such an extension. With this approach Semantic-OGSA semantically enables basic OGSA services.

The Registry is another key architectural element. A decentralized hierarchical structure is well suited to the e-Government domain where some kind of hierarchical topology of organizations (national, regional, local) often exists.

We hence started our studies by investigating UDDI 3.0 registries federation. Since we intend to use federated UDDI registries to discovery both WS and WS-Resources we provide a mechanism to “refresh” resource status across the whole domain: whenever the status of a specific resource changes, the federated registry structure as a whole must be aware of it. To obtain a registry able to manage all the needed information, it is necessary to extend the registry with metadata annotations. We will follow this approach to add semantic Web capabilities to UDDI registries [14][15].

4 Scenario

In a crisis, several organizations work together as a virtual organization, sharing resources across organizational boundaries to deal with the complexities of such situations. In those scenarios resources are mainly physical ones: police cars, ambulances, emergency professionals and volunteers. Disaster response VO are characterized by resource heterogeneity and must rapidly reconfigure (structural and functional changes) to adapt to the changing communication and control demands present during crisis events [3]. All this requires dynamic and adaptive workflows, able to coordinate fixed and mobile resources on the basis of their readiness, availability and capabilities.

In our scenario, local Emergency Operations Centers (ECOs) are in charge of collect information and coordinating operations. In order to facilitate communication between VO members (EOC chief, workers and volunteer), our solution provides multimodal interaction support to enabled devices of the mobile operators involved in response operations, exploiting a “point and sketch” interaction mode, which is particularly useful in on-field mobile operation.

In case of emergency, mobile social community users, are asked by a Resource Planning Support (RPS) service to be involved in the operation and eventually integrated into an *ad-hoc* emergency VO created by the EOC. In this way, we increase on-field operator team with unexpected and unplanned units (resources). Furthermore, accessing community member profiles, the RPS service can organize operations considering user’s skills and assigning the best task to each VO member.

Let’s suppose, for example, that after an earthquake some teams are involved in on-site damage control on the affected area, may be requesting assistance.

Imagine that some team’s member with medical skill identifies the symptoms of an heart attack for a citizen asking help. He can use his multimodal mobile device to request a properly equipped ambulance in the place that he points-to on the screen

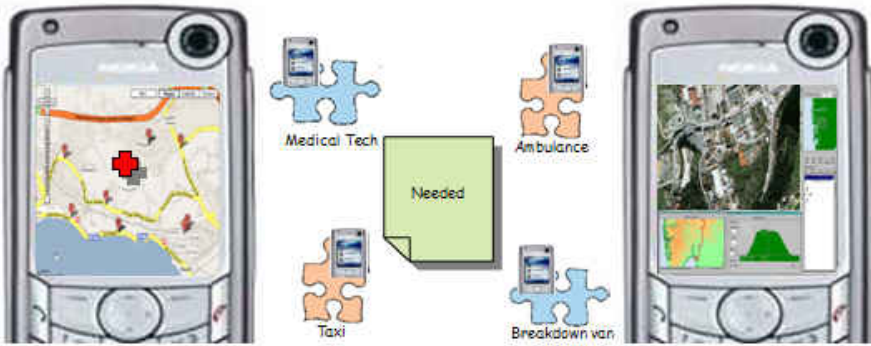


Fig. 4. Client application : collecting useful information



Fig. 5. Server side : information management

map. Through support services, every organization in the VO that can provide ambulance resources, will be asked to return availability, location, capabilities (equipment and crew) and other relevant information.

The available resources will be discovered on the (federated) registry and the one best matching the need will be called to accomplish the task.

5 Conclusion and Future Work

So far the research activities already carried out show that the described approach is feasible, although it places high demands (in terms of computing power) to back end systems.

The availability of suitable distributed input processing software for voice recognition on large scale is still elusive.

Future research activities will deal with solving those aspects and improving the processing of context sensitive but user unaware input. For example in case of an

unexpected raise in temperature an environment sensor may signal the risk of a fire even if the user does not recognize visually a fire.

References

- [1] NEM - Strategic Research Agenda, Version 4.0 (August 2006)
- [2] Lugano, G.: Mobile Social Software: Definition, Scope and Applications. In: EU/IST eChallenges Conference, The Hague, The Netherlands (2007)
- [3] Mehrotra, S., Znati, T., Thompson, C.W.: Crisis Management. IEEE Internet Computing Magazine (June/February 2008)
- [4] Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International J. Supercomputer Applications* 15(3) (2001)
- [5] The UN E-Government Survey 2008: From E-Government to Connected Governance, United Nation Public Administration Network (2008), http://www2.unpan.org/egovkb/global_reports/08report.htm
- [6] Foster, I., Kishimoto, H., Savva, A., Berry, D., Djaoui, A., Grimshaw, A., Horn, B., Maciel, F., Siebenlist, F., Subramaniam, R., Treadwell, J., Von Reich, J.: The Open Grid Services Architecture, Version 1.5, Open Grid Forum, Lemont, Illinois, U.S.A, GFD-I.080 (September 2006)
- [7] Web Services Resource Framework 1.2 TC. OASIS (April 2006)
- [8] Foster, I., Frey, J., Graham, S., Tuecke, S., Czajkowski, K., Ferguson, D., Leymann, F., Nally, M., Sedukhin, I., Snelling, D., Storey, T., Vambenepe, W., Weerawarana, S.: Modeling Statefull Resources with Web Services v.1.1
- [9] Corcho, O., Alper, P., Kotsiopoulos, I., Missier, P., Bechhofer, S., Goble, C.: An overview of S-OGSA: A Reference Semantic Grid Architecture's Web Semantics. *Science, Services and Agents on the World Wide Web* 4(2) (June 2006)
- [10] Web Services Business Process Execution Language, OASIS, http://www.oasisopen.org/committees/tc_home.php?wg_abbrev=wsbpel
- [11] Dörnemann, T., Friese, T., Herdt, S., Juhnke, E., Freisleben, B.: Grid Workflow Modelling Using Grid-Specific BPEL Extensions, *German e-Science 2007* (2007)
- [12] CNIPA, Sistema Pubblico di Cooperazione: Quadro Tecnico d'Insieme ver 1.0 (October 14, 2005), http://www.cnipa.gov.it/site/_files/SPCoop-QuadroInsieme_v1%200_20051014.pdf
- [13] CNIPA, Sistema Pubblico di Cooperazione: Termini e Definizioni (October 14, 2005), http://www.cnipa.gov.it/site/_files/SPCoop-TerminiDefinizioni_v1.0_20051014.pdf
- [14] Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.P.: Semantic Matching of Web Services Capabilities. In: *International Semantic Web Conference 2002*, pp. 333–347 (2002)
- [15] Paolucci, M., Kawamura, T., Payne, T., Sycara, K.: Importing the SemanticWeb in UDDI. In: *Web Services, E-Business and Semantic Web Workshop* (2002)
- [16] European Interoperability Framework for pan-European e-Government Services version 1.0, <http://europa.eu.int/idabc/en/document/3761>
- [17] The future of e-Government: an exploration of ICT-driven models of e-Government for the EU in 2020, <http://ipts.jrc.ec.europa.eu/publications/pub.cfm?id=1481>

- [18] I.D.A.B.C., Interoperable Delivery of European e-Government Services to public Administrations, Businesses and Citizens, <http://ec.europa.eu/idabc/en/document/5101>
- [19] Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. *Scientific American Magazine* (May 17, 2001) (retrieved March 26, 2008)
- [20] Reference Model for Service Oriented Architecture 1.0, OASIS (August 2006)
- [21] Little, M., Webber, J., Parastatidis, S.: Stateful Interactions in Web Services. A comparison of WSContext and WS-Resource Framework. *SOA World Magazine* (April 2004)
- [22] W3C Multimodal Interaction Activity, <http://www.w3.org/2002/mmi/>

A Comparison of Pseudo-paper and Paper Prototyping Methods for Mobile Evaluations

Joanna Lumsden and Ryan MacLean

National Research Council of Canada, IIT e-Business, 46 Dineen Drive, Fredericton, N.B.,
Canada E3B 9W4
jo.lumsden@nrc-cnrc.gc.ca

Abstract. The research presented in this paper is part of an ongoing investigation into how best to support *meaningful* lab-based usability evaluations of mobile technologies. In particular, we report on a comparative study of (a) a standard paper prototype of a mobile application used to perform an early-phase seated (static) usability evaluation, and (b) a pseudo-paper prototype *created from the paper prototype* used to perform an early-phase, *contextually-relevant, mobile* usability evaluation. We draw some initial conclusions regarding whether it is worth the added effort of conducting a usability evaluation of a pseudo-paper prototype in a contextually-relevant setting during early-phase user interface development.

Keywords: mobile technology, paper prototyping, mobile lab-based usability evaluations.

1 Introduction

The benefits of paper prototyping are well recognized for desktop system design. Virzi *et al.* [1] compared the number of usability problems found via a think-aloud protocol using a low-fidelity prototype (a paper prototype) to the number found using a high-fidelity prototype (a final product) of an electronic encyclopedia. They found relatively little difference with respect to the number of usability problems uncovered using the two different prototypes; their findings suggest that a low-fidelity prototype is just as capable of finding usability problems as a high-fidelity prototype at a comparable degree of sensitivity. In a similar study, Sefelin *et al.* [2] compared a low-fidelity *paper* prototype to a low-fidelity *computer-based* prototype in order to determine whether participants in a usability study would be more likely to critique one or the other. They found that the prototype medium was relatively unrelated to the willingness of participants to make critical suggestions about the system design.

Less is known, however, about the benefits of *paper* prototyping for evaluation of *mobile* application designs [3]. Mobile systems are typically used by people who are *mobile*, in dynamically changing, contextually-rich and complex environments; many of the usability problems within mobile application designs are, therefore, best discovered through evaluation of the system in environments representative of the real world [e.g., 4, 5, 6]. Although mobile system design could potentially benefit from early-stage usability studies based on paper prototypes, such studies are rarely performed due to challenges presented by the *mobile* use of such prototypes [3].

Hendry *et al.* [7] created a paper prototype of a mobile application using a cardboard box the size of a table PC and laminated cards to simulate the screens of the user interface. To test their design with their target users, Hendry *et al.* conducted an ‘on the street’ field trial. Although they recognized the immense benefit in using early-stage prototypes with target users in situ, they reported great difficulty in using the prototype in the field: although participants were seated when using the prototype, and so held the box on their laps, they were forced to keep all the other prototype components on the ground or in pockets and, as a result, they found that the prevailing wind was a particular nuisance.

Sá and Carriço [3] discuss their experiences with low-fidelity prototypes for two different mobile applications. They initially constructed their prototypes out of card and Post-It notes, but observed that the dimension, weight, and handling of these truly paper prototypes “mised” participants about the form factor of the final products. They also found their paper prototypes to be too fragile; they weakened in structural integrity when used in the same manner as a final product (e.g., when placed in pockets, etc.). To address these concerns, Sá and Carriço created wooden frames that approximated the size, shape, and configuration of the final devices; these were then used to hold “screen cards”. They found that this solution not only proved durable, but it also allowed users to comment on the shape of the device and the placement of buttons relative to how they held the device.

The research presented in this paper represents an *initial* investigation into the comparative strengths of a traditional paper prototype used in a seated evaluation protocol to a pseudo-paper prototype (created from the paper prototype) used in a mobile, lab-based protocol, in terms of the number and severity of usability problems identified using each. This study draws on research in the field of *effective* mobile, *lab-based* usability evaluation [e.g., 6, 8, 9] to expand on the findings of Sá and Carriço [3] and Hendry *et al.* [7] in order to further our understanding of potential mechanisms by which to effectively (and conveniently) use paper prototypes in *lab-based*, *mobile* evaluations, as well as to discover the relative merits of doing so (as compared to simply employing a traditional seated protocol). The following sections describe our evaluation design and process, and discuss our results, respectively. We conclude, in Section 4, with a brief discussion of further work.

2 Evaluation Design and Process

Our study was based on a paper prototype of a mobile system designed to be used in a grocery store to enhance the shopping experience. It was, in essence, a shopping cart-mounted, shop-and-scan system designed to support consumers’ choice of products based on health attributes, price, or customer ratings. Figure 1 shows a sample screen from the paper prototype.

The dimensions of the paper prototype reflect the actual screen size of the end device (a Fujitsu tablet) so as to prevent misleading users on issues of screen dimension. Specific screen mock-ups were developed based on two shopping scenarios which we used in our study.



Fig. 1. Sample screen from the paper prototype

Taking as inspiration a tool developed by Sá and Carriço [3], we developed a *pseudo*-paper prototype of our application based on the paper prototype shown in Figure 1. We took a series of digital photographs of the paper prototype at each stage during a walkthrough of our two shopping scenarios; we then organized these photographs into a PowerPoint presentation, linking them together by creating invisible clickable areas over the prototype's buttons – thus allowing participants to progress through the various screens associated with our study scenarios. We installed our pseudo-paper prototype on our Fujitsu tablet; thus participants could interact with the clickable areas by tapping the touchscreen of the device. To handle system response where a user 'scanned' a product (i.e., where interaction would be focused on a scanner rather than the touchscreen), we developed a secondary wireless application to allow us to remotely advance the PowerPoint slide when we saw a user 'scanning' a product.

For manageability, we did not 'activate' components of the photographs that would have constituted incorrect user actions relative to our scenarios; during our evaluations, we were able to observe such action intentions, but our pseudo-prototype limited users' ability to follow through on such intentions. Similarly, after piloting the use of the paper prototype (with the complexity of individual paper components for each screen) we decided to work with printouts of the same digital photographs used for our pseudo-paper prototype, and to verbally limit activation of 'misguided' user intentions. Although we recognize the limitations of this approach in terms of curtailing exploratory behavior and preventing, to some extent, observation of recovery behavior, we feel our decision placed both prototypes on an even basis for the purpose of our study – namely, to compare the impact of the *evaluation protocols* rather than conduct an in-depth evaluation of our specific design; furthermore, we eliminated, for the paper prototype, some of the lag in 'screen updates' brought about by the need for



Fig. 2. Participant using the pseudo-paper prototype in contextually-rich, lab-based study

the evaluator to manually reset the prototype after each user action, thereby bringing it more in line with the pseudo-paper prototype in this regard.

Appropriately designed lab-based studies have proven a viable means by which to meaningfully assess the usability of mobile applications under controlled, experimental conditions [8, 9]. For the purpose of evaluating our pseudo-paper prototype, we therefore designed our study to reflect (albeit, abstractly) realistic environmental conditions – namely, a grocery store.

Figure 2 shows our experimental set-up for our pseudo-paper prototype. We mounted the tablet onto a ‘shopping cart’ which participants were required to navigate around a series of ‘aisles’ in order to select and scan products (props that were attached to our ‘aisles’) based on provided shopping lists/scenarios. The evaluator followed each participant in order to determine when a product was scanned, and to subsequently progress the PowerPoint slide as applicable. As participants were completing their study tasks, they were surrounded by ambient grocery store sounds [10] played at $\sim 69\text{dB(A)}$ (based on real world readings we had taken previously).

Our study design for the paper prototype simply required participants to interact with the paper version of the prototype whilst seated at a desk; the evaluator ‘acted’ as the computer, manipulating the components of the prototype in response to participants’ actions. To bring parity to the two studies, we used the same ambient grocery store noise in this study set-up; additionally, we provided participants with the product props for the products itemized in the shopping lists/scenarios, albeit they were just available on the table next to the participants.

We adopted a between-groups design, assigning participants to one of two groups based on prototype. Participants in each group were given minimal training in the use of the system since it was designed to be ‘walk-up-and-usable’ without training. All participants were required to work through the same two provided shopping scenarios, the order of completion being counterbalanced across participants in each group. Across both groups, we used a think-aloud protocol combined with audio/video

recordings of users' commentary and actions. Twelve people participated in our study, six per prototype/group.

3 Results and Discussion

We generated a content log of participants' activities and commentaries based on the audio/video recordings for each session. We applied two qualitative analysis techniques to the content logs. In the first instance, we conducted a usability defect analysis to compare the types and distribution of usability defects identified using the two prototypes; we then performed a heuristic analysis based on the content log data to determine the uniqueness and severity of problems found using each prototype. The following sections reflect on our analyses.

3.1 Usability Defect Analysis

Each content log was analyzed to identify and tag usability problems according to Lindgaard's [11] categorization of usability defects. Figure 3 lists the defect categories, and shows the number of instances of each according to prototype.

In *general*, the distribution of usability defects follows a similar pattern across the two prototypes. Two exceptions to this lie in the *Screen Design & Layout* and *Terminology* categories; while, for the paper prototype, we see a drop in the number of usability defects in these two categories compared to the *Navigation* category, we see the opposite for the pseudo-paper prototype. Both the *Screen Design & Layout* and *Terminology* categories relate directly to how quickly users can find and recognize user interface elements. A possible explanation for the divergence across the two prototypes for these categories could be that, on account of the fact that they were seated and not required to multitask, participants using the paper prototype may have benefited from an increased capacity, or felt it more appropriate, to scrutinize these aspects across the whole design; conversely, on account of the fact that they were required to multitask, participants using the pseudo-paper prototype likely only paid direct attention to *specific* aspects of the layout and terminology of the interface as/when they were needed. Additionally, given the contextual relevance of the study protocol, participants using the pseudo-paper prototype may have felt it contextually inappropriate to clinically 'examine' the design, but instead felt compelled to simply use the system much as they would in the real world. This suggests that the results from the pseudo-paper prototype might better reflect the realistic 'walk-up-and-usability' of the design.

Across all defect categories, the number of instances identified by participants using the pseudo-paper prototype is equal to or (often substantially) greater than the number identified by participants using the paper prototype (see Figure 3); in total, 251 defects were identified using the former, compared to 195 identified using the latter). This difference is particularly noticeable in the number of *Terminology* and *Match with User Tasks* defects identified using each prototype. We suggest that the substantial increase in number of defects identified in these categories by participants using the pseudo-paper prototype demonstrates the potential impact of the contextually-relevant, mobile setting in which the prototype was evaluated. That is, when

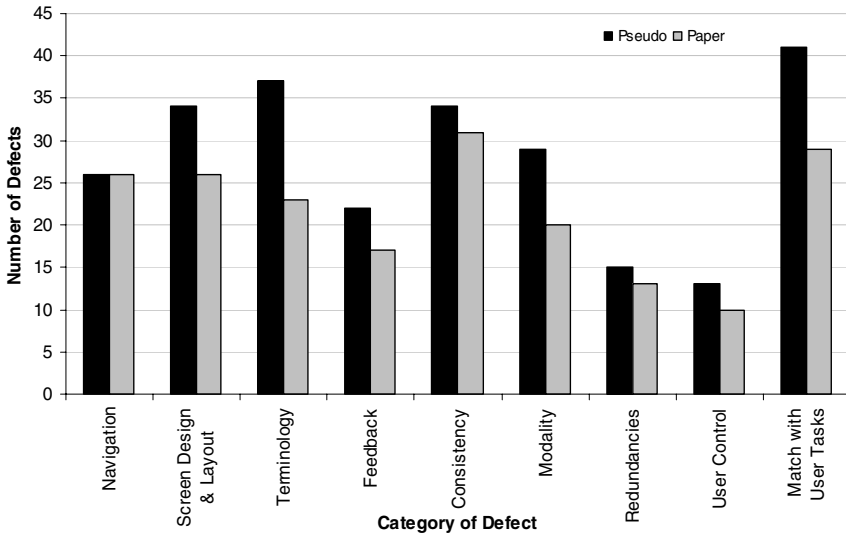


Fig. 3. Total number of usability issues identified according to category and prototype

participants were required to interact with the system in a setting that (a) reflected the need to actively multitask and (b) the cognitive challenges associated with shopping in the real world, the inappropriateness of terminological aspects of the design, as well as the limited degree to which the design fitted with the primary tasks of the user, became obvious; the use of the paper prototype did not support such extensive observations. We would suggest that the use of the pseudo-paper prototype within a contextually-relevant, mobile protocol has proven noticeably more effective at enabling us to identify usability defects in the design, and to highlight where the bulk of defects lie with respect to realistic usage scenarios.

3.2 Heuristic Analysis

Each content log was again analyzed, this time to identify and tag usability problems according to the mobile heuristics appropriated by Bertini *et al.* [4], as summarized in Table 1. Like most usability heuristics, Bertini *et al.*'s mobile heuristics are intended to be used by usability experts during direct, hands-on analysis of a user interface design. By applying the heuristics to the *observed* interactions of *test users*, we appreciate that our use of the heuristics is slightly unorthodox, but we simply used them to provide an alternative means by which to classify the usability problems we observed. Given the typical complexity of context of use of mobile applications (as highlighted in our case study) it is imperative that the heuristics proposed by Bertini *et al.* are observed if true mobile usability is to be achieved; as such, we felt these heuristics provided a good comparative measure of the efficacy of our evaluation protocols.

Table 1. Summary of mobile usability heuristics

Heuristic	Description
1	Visibility of system status & losabilty/findability of the mobile device
2	Match between system and the real world
3	Consistency and mapping
4	Good ergonomics & minimalist design
5	Ease of input, screen readability, and glancability
6	Flexibility, efficiency of use, and personalization
7	Aesthetic, privacy, and social conventions
8	Realistic error management

Figure 4 shows the distribution (or nature) of *unique* usability problems found using each of the prototypes, as well as the unique instances of common (overlapping) usability problems found using both. Overall, participants using the pseudo-paper prototype found the most unique usability problems (27); in contrast, participants using the paper prototype found 19 unique problems, with an additional 19 problems being found irrespective of prototype.

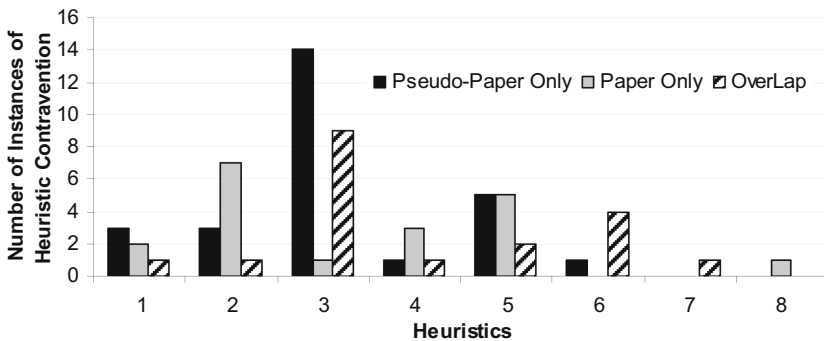


Fig. 4. Distribution of unique usability problems according to heuristic and prototype (including problems identified using both prototypes – overlap)

As can be seen from Figure 4, there were more instances of mismatch between the user interface design and real world context (Heuristic 2) identified using the paper prototype than the pseudo-paper prototype. We suggest (on the basis of user comments and our observations) that this was because, by using the pseudo-paper prototype in a contextually-relevant, mobile setting, participants were better able to match aspects of the system to the context in which the system was designed to be used; conversely, seated participants using the paper prototype were forced to ‘imagine’ the context of use, and as such often over, or inappropriately, analyzed the situation.

Although there was considerable overlap in the defects found according to Heuristic 3 (*consistency and mapping*), there was also a large gap in the number of such usability problems identified via the use of the pseudo-paper prototype compared to the paper prototype. The focus of Heuristic 3 is closely related to, and concurs with the findings for, the *Screen Design & Layout*, *Terminology*, and *Match with User Tasks* defect categories discussed previously: consistency and mapping are concerned

with how participants *think* things should work, and this internalized mapping is based, in large part, on the clarity of screen design, including terminology used.

Neither prototype led to identification of many *aesthetic, privacy, and social convention* problems (Heuristic 7); this is unsurprising, given that both prototypes were being used in a lab without other people around. Whilst there is scope to remedy this in the mobile, contextually-relevant evaluation protocol supported by the pseudo-paper prototype, it is doubtful this could be addressed using the paper prototype.

Each identified and classified usability problem was additionally given a severity rating based on Nielsen's Severity Rating Scale, as used by Bertini *et al.* [4], namely: cosmetic; minor; major; and catastrophic. Figure 4 shows the breakdown of severity of the unique, heuristically-derived usability problems according to prototype.

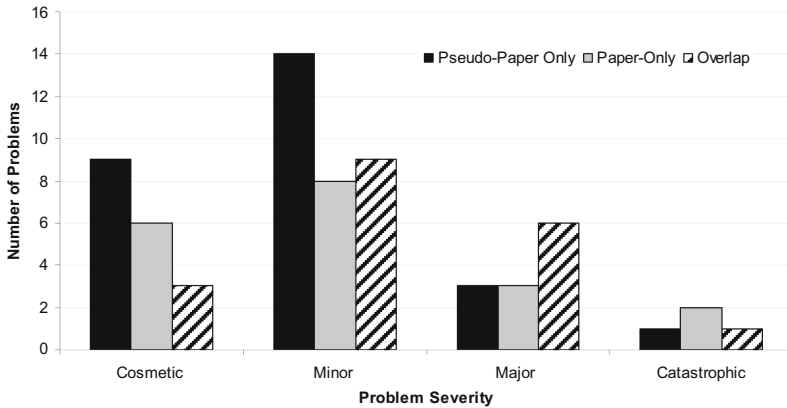


Fig. 5. Severity of problems according to prototype

Encouragingly, the use of the pseudo-paper prototype not only identified more usability problems, but the problems identified were distributed across all 4 levels of severity; with the exception of *catastrophic* problems, the relative distribution of problem severity was similar for both prototypes. Participants using the paper prototype identified two unique catastrophic problems that were not noted by participants using the pseudo-paper prototype; on closer inspection of the specifics of these problems, however, we noted that they related to knowing how to ‘check-out’ and handle fruit and vegetables – neither of which were part of the evaluation scenario, and neither of which were a noted issue when the contextual-relevance of the pseudo-paper prototype environment was present. As a consequence, in our case, we do not see the lower extent of identification of catastrophic problems when using the pseudo-paper prototype to be a drawback of the prototype, but rather a reflection of its ability to mediate the severity of problems relative to context.

4 Conclusions and Further Work

We attribute no statistical significance to the observations we have presented in this paper; instead, we present our results as an *initial observation* of the differences in

usability problems identified using a paper prototype of a mobile application in a traditional static evaluation setting versus a pseudo-paper prototype of the same application in a contextually-relevant mobile evaluation setting (that is, taking advantage of the contextual relevance the pseudo format could support in its respective evaluation protocol). We have shown that the use of the pseudo-paper prototype allows participants to identify more usability problems (whilst maintaining a similar distribution of defects to that observed with the paper prototype), to identify more unique usability problems (again, preserving distribution), and to be compatible with the paper prototype in terms of supporting the identification of usability problems across the various severity levels. We have also shown the benefits of a pseudo-paper prototype in terms of its ability to be used within a contextually-relevant experimental protocol, such that the problems identified better reflect what might happen in real use – that is, we consider the results more *meaningful*.

We have, obviously, only compared the paper and pseudo-paper prototypes relative to one application domain. We feel it would be beneficial to repeat the evaluation for additional application domains to determine the generalizability of our findings. Once in possession of such data, we would then be in a position to conduct deeper, statistical analysis to further demonstrate the merit of conducting *contextually-rich, mobile* usability evaluations of *pseudo*-paper prototypes in the early-phases of mobile UI design. We also anticipate comparing the benefits of our *pseudo*-paper prototyping approach to other, increasingly established, mechanisms for evaluation of mobile user interface designs. To conclude, therefore, we present, here, the results of our existing qualitative analysis as a *first* indication of the potential usefulness of such an approach in the hope that developers can begin to benefit from this early investigation.

References

1. Virzi, R.A., Sokolov, J.L., Karis, D.: Usability Problem Identification Using Both Low- and High-Fidelity Prototypes. In: Proceedings of SIGHCI Conference on Human Factors in Computing Systems (CHI 1996), Vancouver, Canada, April 13-18, pp. 236–243 (1996)
2. Sefelin, R., Tscheligi, M., Giller, V.: Paper Prototyping - What is it Good For? A Comparison of Paper- and Computer-Based Low-Fidelity Prototyping. In: Proceedings of Conference on Human Factors in Computing Systems - Extended Abstracts (CHI 2003), Ft. Lauderdale, USA, April 5-10, pp. 778–779 (2003)
3. Sá, M., Carriço, L.: Low-Fi Prototyping for Mobile Devices. In: Proceedings of SIGCHI Conference on Human Factors in Computing Systems (CHI 2006), April 22-27, pp. 694–699 (2006)
4. Bertini, E., Gabrielli, S., Kimani, S.: Appropriating and Assessing Heuristics for Mobile Computing. In: Proceedings of Workshop on Advised Visual Interfaces (AVI 2006), Venezia, Italy, May 23-26, pp. 119–126 (2006)
5. Kjeldskov, J., Graham, C., Pedell, S., Vetere, F., Howards, S., Balbo, S., Davies, J.: Evaluating the Usability of a Mobile Guide. The Influence of Location, Participants, and Resources. *Behaviour and Information Technology* 24(1), 51–65 (2005)
6. Kjeldskov, J., Stage, J.: New Techniques for Usability Evaluation of Mobile Systems. *International Journal of Human Computer Studies (IJHCS)* 60(5-6), 599–620 (2004)

7. Hendry, D.G., Mackenzie, S., Kurth, A., Spielberg, F., Larkin, J.: Evaluating Paper Prototypes on the Street. In: Proceedings of Conference on Human Factors in Computing Systems - Extended Abstracts (CHI 2005), Portland, USA, April 2-7, pp. 1447-1450 (2005)
8. Kjeldskov, J., Skov, M.B., Als, B.S., Høegh, R.T.: Is It Worth the Hassle? Exploring the Added Value of Evaluating the Usability of Context-Aware Mobile Systems in the Field. In: Proceedings of 6th International Symposium on Mobile Human-Computer Interaction (MobileHCI 2004), Glasgow, Scotland, September 13 - 16, pp. 61-73 (2004)
9. Lumsden, J., Kondratova, I., Langton, N.: Bringing A Construction Site Into The Lab: A Context-Relevant Lab-Based Evaluation Of A Multimodal Mobile Application. In: Proceedings of 1st International Workshop on Multimodal and Pervasive Services (MAPS 2006), Lyon, France, June 29, pp. 62-68 (2006)
10. PacDV, Grocery Store,
http://www.pacdv.com/sounds/ambience_sounds.html
11. Lindgaard, G.: Usability Testing & System Evaluation: A Guide for Designing Useful Computer Systems, 1st edn. Chapman & Hall, London (1994)

A Model for Checking the Integrity Constraints of Mobile Databases

Hamidah Ibrahim¹, Zarina Dzolkhifli¹, and Praveen Madiraju²

¹ Department of Computer Science
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia
hamidah@fsktm.upm.edu.my

² Department of Mathematics, Statistics, and Computer Science
Marquette University, USA
praveen@mscs.mu.edu

Abstract. In this paper we propose Three-Level (3-L) model, wherein the process of constraint checking to maintain the consistent state of mobile databases is realized at three different levels. Sufficient and complete tests proposed in the previous works together with the idea of caching relevant data items for checking the integrity constraints are adopted. This has improved the checking mechanism by preventing delays during the process of checking constraints and performing the update. Also, the 3-L model reduces the amount of data accessed given that much of the tasks are performed at the mobile host, and hence speeds up the checking process.

Keywords: Mobile databases, integrity constraints, constraint checking.

1 Introduction

Recently, there has been an increasing interest in mobile computing due to the rapid advances in wireless communication and portable computing technologies. Massive research efforts from academia and industry have been put forth to support a new class of mobile applications such as just-in-time stock trading, mobile health services, mobile commerce, and mobile games as well as migrating the normal conventional applications to mobile applications. Users of these applications can access information at any place at any time via mobile computers and devices such as mobile phone, palmtops, laptops, and PDA [7].

While technology has been rapidly advancing, various constraints inherited from limitations of wireless communication and mobile devices remain primary challenges in the design and implementation of mobile systems and applications. These constraints include: limited client capability, limited bandwidth, weak connectivity and user mobility. Mobile devices generally have poor resources and thus it is usually impossible for them to store all data items in the network. In addition, disconnections occur frequently, which may be intentional or unintentional. These constraints make the wireless and mobile computing environments uniquely different from a conventional wired server/client environment [7].

A general architecture of a mobile environment consists of base stations (BS) and mobile hosts (MH). The base station is a stationary component in the model and is responsible for a small geographic area called a cell. They are connected to each other through fixed networks. The mobile host is the mobile component of the model and may move from one cell to another. These mobile hosts communicate with the base stations through wireless networks.

Since a mobile host is not capable of storing all data items in the network, thus it must share some data item with a database in the fixed network. Any update operation or transaction that occurs at the mobile host must guarantee *database consistency*. A database state is said to be consistent if the database satisfies a set of statements, called *integrity constraints*, which specify those configurations of the data that are considered semantically correct. The process of ensuring that the integrity constraints are satisfied by the database after it has been updated is termed *constraint checking*, which generally involves the execution of *integrity tests*. In a mobile environment, checking the integrity constraints to ensure the correctness of the database spans at least the mobile host and one other database (node), and thus the update is no longer local but rather distributed [10]. As mentioned in [10], the major problem in the mobile environment are unbounded and unpredictable delays can affect not only the update but other updates running at both the mobile and the base stations, which is clearly not acceptable for most applications. With the same intuition as [10], we address the challenge of extending the data consistency maintenance to cover disconnected and mobile operations.

In this paper, a model called the Three-Level (3-L) is proposed where checking the consistency of mobile databases is performed at three different levels. This model is suitable for both intentional (planned) and unintentional (unplanned) disconnection. This model differs from the approach proposed in [10] since it is intended to cater for the important and frequently used integrity constraints, i.e. those that are used in database application. Mazumdar's approach [10] is restricted to set-based constraints (equality and inequality constraints). In our work, in order not to delay the process of checking constraints during disconnection, a similar concept as proposed in distributed databases [6] is employed, namely localizing integrity checking by adopting sufficient and complete tests. Since sufficient test can only verify if a constraint is satisfied, thus we propose that the data items required for the checking to be cached at the mobile host during the relocation period. Our model not only treats the issue of disconnection but also reduces the amount of data accessed during the process of checking the consistency of the mobile databases. Hence, we achieve speed up in the constraint checking process.

This paper is organized as follows. In Section 2, the previous works related to this research are presented. In Section 3, the basic definitions, notations and examples, which are used in the rest of the paper, are set out. Section 4 describes the Three-Level (3-L) model while Section 5 discusses the performance of the Three-Level (3-L) model. Conclusions and further research are presented in the final section, 6.

2 Related Work

Much of the research concerning integrity constraint checking has been conducted in the area of relational database systems. A comprehensive survey on the issues of

constraint checking and maintaining in centralized, distributed and parallel databases is provided in [5]. A naïve approach is to perform the update and then check whether the integrity constraints are satisfied in the new database state. This method, termed *brute force checking*, is very expensive, impractical and can lead to prohibitive processing costs because the evaluation of integrity constraints requires large amounts of data, which are not involved in the database update transition. Hence, improvements to this approach have been reported in many research papers. Many approaches have been proposed for constructing efficient integrity tests, for a given integrity constraint and its relevant update operation, but these approaches are mostly designed for a centralized environment [9]. As centralized environment has only a single site, the approaches concentrate on improving the checking mechanism by minimizing the amount of data to be accessed during the checking process. Hence, these methods are not suitable for mobile environment as the checking process often spans multiple nodes and involves the transfer of data across the network.

Although there are a few studies that have been conducted to improve the checking mechanism by reducing the amount of data transferred across the network in distributed databases such as [1, 3, 6, 8], but these approaches are not suitable for mobile databases. These approaches reformulate the global constraints into local constraints (local tests) with an implicit assumption that all sites are available, which is not true in mobile environment, where a mobile unit may be disconnected for long periods. Even though failure is considered in the distributed environment, but none of the approach cater failure at the node where the update is being executed, i.e. disconnection at the target site. Nevertheless, the localization concept proposed in distributed databases is used in our approach.

Other approaches such as [4, 11] focus on the problems of checking integrity constraints in parallel databases. These approaches are not suitable for mobile databases as the intention of their approach is to speed up the checking process by performing the checking concurrently at several nodes. To the best of our knowledge, PRO-MOTION [10] is the only work that addresses the issues of checking integrity constraints in mobile databases. The difference between our work and the work in [10] has been highlighted in the previous section.

3 Preliminaries

Our approach has been developed in the context of relational databases. Database integrity constraints are expressed in prenex conjunctive normal form with the range restricted property. Integrity tests can be classified into several categories depending on the characteristics of the tests. Three different types of integrity test based on its properties were defined by McCarroll [11], namely: *sufficient tests*, *necessary tests* and *complete tests*. An integrity test has the sufficiency property if when the test is satisfied, this implies that the associated constraint is satisfied and thus the update operation is safe with respect to the constraint. An integrity test has the necessity property if when the test is not satisfied, the associated constraint is violated and thus the update operation is unsafe with respect to the constraint. An integrity test has the completeness property if the test has both the sufficiency and the necessity properties.

Throughout this paper, the following symbols and their intended meaning, which are related to integrity constraints, are used:

- $I^v = \{I_1, I_2, \dots, I_M\}$, the set of integrity constraints of an application in the whole mobile system.
- $I^{Bi} = \{I^{Bi}_1, I^{Bi}_2, \dots, I^{Bi}_N\}$, the set of integrity constraints at the base station, i .
- $I^{Mh} = \{I^{Mh}_1, I^{Mh}_2, \dots, I^{Mh}_O\}$, the set of integrity constraints at the mobile host, h .
- $T_i = \{T_{i1}, T_{i2}, \dots, T_{iw}\}$, the set of integrity tests for a given constraint I_i of I^v .

From the above, $(\cup_{i=1}^P I^{Bi}) \cup (\cup_{h=1}^Q I^{Mh}) = I^v$, where P and Q are the number of base stations and mobile hosts, respectively in the mobile system.

Similarly, the following are the symbols and their intended meaning that are related to the data items in the mobile system. Here, data item refers to relation or fragments of relations that appear in the specification of an update operation.

- $R^v = \{R_1, R_2, \dots, R_S\}$, the set of relations or fragments of relations in the mobile system.
- $R^{Bi} = \{R^{Bi}_1, R^{Bi}_2, \dots, R^{Bi}_T\}$, the set of relations or fragments of relations at the base station, i .
- $R^{Mh} = \{R^{Mh}_1, R^{Mh}_2, \dots, R^{Mh}_U\}$, the set of relations or fragments of relations at the mobile host, h .

From the above, $(\cup_{i=1}^P R^{Bi}) \cup (\cup_{h=1}^Q R^{Mh}) = R^v$, where P and Q are the number of base stations and mobile hosts, respectively in the mobile system. Also, we assume that for each data item, $R^{Mh}_v \in R^{Mh}$, the same data item appears in one of the base station, i.e. $R^{Mh}_v \in (\cup_{i=1}^P R^{Bi})$ [2].

Update operation in a mobile environment can occur at two different levels:

- $U^{Bi}(R)$, an update operation over the relation R , submitted by a user at the base station, i . This type of update operation is not considered in this paper, as this is similar to the update operation in distributed databases. Note that R can also be a fragment of relation.
- $U^{Mh}(R)$, an update operation over the relation R , submitted by a user through his mobile host, h , where R is located at the mobile host. Note also that R can be a fragment of relation.

The symbol, $C(R)$, is used to denote the list of relations, R that occurs in the specification of a construct, C , where C can be an update operation, an integrity constraint or an integrity test. Throughout this paper the *company* database is used, as given in Figure 1. Due to space limitation, only referential and general semantic integrity constraints are used in the examples. Table 1 presents some of the integrity tests generated based on the set of integrity constraints given in Figure 1. The derivation of the integrity tests is omitted here since this is not the focus of this paper. Interested readers may refer to [6].

4 The Three-Level (3-L) Model

As mentioned earlier, this research proposes a model, called the Three-Level (3-L) model to ensure that the consistency of mobile databases is maintained. As the name implies, the model consists of three distinct levels, as depicted in Figure 2.

When a user submits an update operation $U^{Mh}(R)$, through a mobile host Mh , the list of constraints, I^{Mh} , at the mobile host is checked. Violation of any of the constraints will abort the operation. Otherwise, if the checking process does not require information from the other sites, then I^{Mh} is said to be satisfied and the update operation is safe to be performed. The second level is invoked if the information stored at the mobile host is not sufficient to validate whether the constraint I^{Mh} is violated or not. At the first level, the process of checking the constraints spans only the mobile host, i.e. local to the mobile host. The type of test suitable for this level is the sufficient test with the existential quantifier since the mobile host has limited capacity and thus the information (relations) stored at the mobile host is limited. Referring to the Table 1, $(\exists t\exists v\exists w)(emp(t, b, v, w))$ is an example of a sufficient test, which check the existence of at least an employee who is currently working in the department b when an insert operation $insert(emp(a, b, c, d))$ is executed. If such information is available at the mobile host, then we conclude that the initial constraint, I_1 , is satisfied. If there is no such information, then further checking needs to be performed. The properties of the sufficient test can be *upgraded* to be similar to the properties of the complete test if all possibilities of values for the required data item are cached to the mobile hosts. For example, referring to the sufficient test $(\exists t\exists v\exists w)(emp(t, b, v, w))$, one notices that comparison is performed against the values of

<i>Schema:</i>	$emp(eno, dno, ejob, esal); dept(dno, dname, mgrno, mgrsal); proj(eno, dno, pno)$
<i>Integrity Constraints:</i>	
	'The <i>dno</i> of every tuple in the <i>emp</i> relation exists in the <i>dept</i> relation'
$I_1:$	$(\forall t\forall u\forall v\forall w\exists x\exists y\exists z)(emp(t, u, v, w) \rightarrow dept(u, x, y, z))$
	'The <i>eno</i> of every tuple in the <i>proj</i> relation exists in the <i>emp</i> relation'
$I_2:$	$(\forall u\forall v\forall w\exists x\exists y\exists z)(proj(u, v, w) \rightarrow emp(u, x, y, z))$
	'Every employee must earn \leq to the manager in the same department'
$I_3:$	$(\forall t\forall u\forall v\forall w\forall x\forall y\forall z)(emp(t, u, v, w) \wedge dept(u, x, y, z) \rightarrow (w \leq z))$
	'Any department that is working on a project P_1 is also working on project P_2 '
$I_4:$	$(\forall x\forall y\exists z)(proj(x, y, P_1) \rightarrow proj(z, y, P_2))$

Fig. 1. The Company static integrity constraints

Table 1. The integrity tests derived based on the integrity constraints listed in figure 1

I^n	Update Template	Integrity Test
I_1	$insert(emp(a, b, c, d))$	1. $(\exists x\exists y\exists z)(dept(b, x, y, z))^1$
	$delete(dept(a, b, c, d))$	2. $(\exists t\exists v\exists w)(emp(t, b, v, w))^2$
I_2	$insert(proj(a, b, c))$	3. $(\forall t\forall v\forall w)(\neg emp(t, a, v, w))^1$
	$delete(emp(a, b, c))$	4. $(\exists x\exists y\exists z)(emp(a, x, y, z))^1$
I_3	$insert(proj(a, b, P1))$	5. $(\exists v\exists w)(proj(a, v, w))^2$
	$delete(emp(a, b, c))$	6. $(\forall v\forall w)(\neg proj(a, v, w))^1$
I_4	$insert(emp(a, b, c, d))$	7. $(\forall x\forall y\forall z)(\neg dept(b, x, y, z) \vee (d \leq z))^1$
	$delete(proj(a, b, P2))$	8. $(\exists t\exists v\exists w)(emp(t, b, v, w) \wedge (w \geq d))^2$
I_4	$insert(proj(a, b, P1))$	9. $(\exists z)(proj(z, b, P2))^1$
	$delete(proj(a, b, P2))$	10. $(\exists z)(proj(z, b, P1))^2$
	$insert(proj(a, b, P1))$	11. $(\forall x)(\neg proj(x, b, P1))^1$
	$delete(proj(a, b, P2))$	12. $(\exists z)(proj(z, b, P2) \wedge (z \neq a))^2$

Note: a, b, c and d are generic constants; ¹: complete test; and ²: sufficient test.

the *dno*. Assuming that the company has four departments, a vertical fragment of the *dept* table consisting of the distinct *dno* is cached to the mobile host, then performing the test against these data items can verify whether the test is satisfied or not, and eventually verify if the initial constraint is satisfied or violated. Caching can be performed during the relocation period.

The second level commences if the mobile host failed to validate the truth of the I^{Mh} . The base station in the current position of the mobile host is responsible for checking the constraints. The base station checks the validity of the constraints against the data stored at its location. At this level, the process of checking the constraints spans the current cell of the mobile host, i.e. local to a cell of the current location of the mobile host. The types of test suitable for this level are the sufficient test and the complete test with the existential quantifier. If the information stored at the base station is not sufficient then the next level is invoked. However, if violation is detected then the base station notifies the mobile host to abort the update operation. The update operation is safe to be performed if no violation is detected.

The next level, third level, spans the remote base station(s), checks the validity of the constraints against the data stored at the remote site(s). Depending on the protocol of the mobile environment, either the flooding technique or the broadcasting technique is used to perform the constraint checking at this level. Here, the types of test that can be used are sufficient as well as complete test.

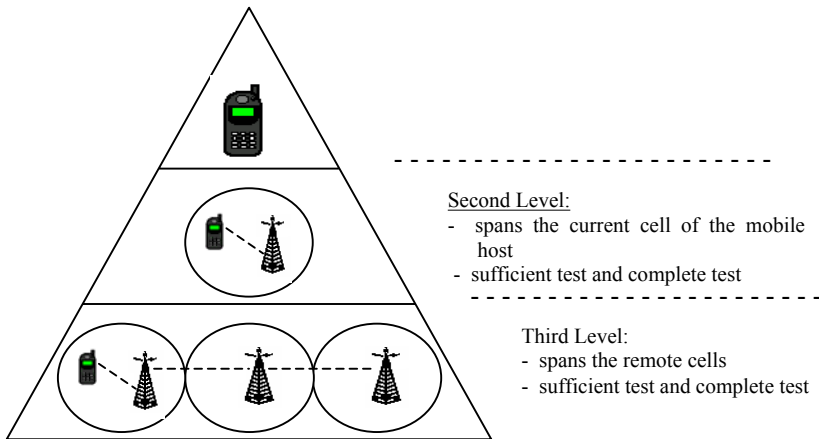


Fig. 2. The Three-Level (3-L) model

Below are the procedures that are employed in the Three-Level (3-L) model and their algorithms (in pseudo code) are presented in Figure 3.

• Procedure *LocateIntegrityConstraints&Tests*(I^v , I^{Mh} , I^{Bi} , T^{Mh} , T^{Bi}) – this procedure is invoked to determine the location of each of the integrity constraints in the I^v . An integrity constraint with its associated integrity tests is located either at the mobile host or base station or both depending on the data items referred in the specification of the constraint and the data items stored at the mobile host and base stations.

- Procedure *CacheDataItem*(T^{Mh} , *DataItem*) – this procedure analyses the list of integrity tests located at the mobile host, T^{Mh} , and cache the required data items to the mobile host. This is performed by the base station of the current cell of the mobile host during the relocation period.
- Procedure *SelectIntegrityConstraints*($U^{Mh}(R)$, I^{Mh} , *Selected- I^{Mh}*) – this procedure is invoked to identify and select only those constraints from the list I^{Mh} , that might be violated given the update operation, $U^{Mh}(R)$. This approach known as the incremental checking has been adopted by many researchers. The procedure is executed at the mobile host.
- Procedure *FirstLevel*($U^{Mh}(R)$, *Selected- I^{Mh}* , *Action*) – this procedure is invoked at the mobile host to check the validity of each of the constraint in the *Selected- I^{Mh}* , given the update operation $U^{Mh}(R)$. Based on the result, an appropriate action is performed.
- Procedure *SecondLevel*(*MHId*, *BSId*, $U^{Mh}(R)$, *SecondLevelI*, *Action*) – this procedure is invoked at the current base station of the mobile host to check the validity of each of the constraint in the *SecondLevelI*, given the update operation $U^{Mh}(R)$. Based on the result, an appropriate action is performed.
- Procedure *ThirdLevel*(*MHId*, *BSId*, $U^{Mh}(R)$, *ThirdLevelI*, *Action*) – this procedure is invoked at the remote base station(s) to check the validity of each of the constraint in the *ThirdLevelI*, given the update operation $U^{Mh}(R)$. Based on the result, an appropriate action is performed.
- Procedure *Notify*(*MHId*, *BSId*, *Action*) – this procedure is invoked at the mobile host to perform the action as indicated by *Action*.

Procedure *LocateIntegrityConstraints&Tests*(I^0 , I^{Mh} , I^{Bi} , T^{Mh} , T^{Bi})

Input: the initial list of integrity constraints, I^0 and its associated integrity tests

Output: the list of integrity constraints I^{Mh} and its associated integrity tests T^{Mh} located at mobile host h and the list of integrity constraints I^{Bi} and its associated integrity tests T^{Bi} located at the base station i

1. Begin
2. $I^{Mh} = \{\}$, $I^{Bi} = \{\}$, $T^{Mh} = \{\}$, $T^{Bi} = \{\}$
3. For each I_j in I^0 do
4. Begin
5. For $h = 1$ to Q do If $I_j(R) \cap R^{Mh} \neq \{\}$ then $I^{Mh} = I^{Mh} \cup I_j$, $T^{Mh} = T^{Mh} \cup T_j$
6. For $i = 1$ to P do If $I_j(R) \cap R^{Bi} \neq \{\}$ then $I^{Bi} = I^{Bi} \cup I_j$, $T^{Bi} = T^{Bi} \cup T_j$
7. End
8. End

Procedure *CacheDataItem*(T^{Mh} , *DataItem*)

Input: the list of integrity tests located at h , T^{Mh}

Output: the list of data items to be cached to h

1. Begin
2. *DataItem* = $\{\}$
3. For each T_j^{Mh} in T^{Mh} do
4. Begin
5. Identify the required data item, D
6. *DataItem* = *DataItem* \cup D
7. End
8. End

Procedure *SelectIntegrityConstraints*($U^{Mh}(R)$, I^{Mh} , *Selected- I^{Mh}*)

Input: the update operation, $U^{Mh}(R)$ and the list of integrity constraints located at h , I^{Mh}

Output: the list of integrity constraints that might violate the $U^{Mh}(R)$, *Selected- I^{Mh}* with its associated integrity tests

1. Begin
2. *Selected- I^{Mh}* = { }
3. For each I^{Mh}_j in I^{Mh} do
4. Begin
5. If the relation and the operation of the update operation $U^{Mh}(R)$ = the relation and operation of the update template (see Table 1 as examples) of the I^{Mh}_j then
6. Begin
7. *Selected- I^{Mh}* = *Selected- I^{Mh}* \cup I^{Mh}_j
8. For each test T_{ji} of I^{Mh}_j do parameterize the test T_{ji} with details of the update $U^{Mh}(R)$
9. End
10. End
11. End

Procedure *FirstLevel*($U^{Mh}(R)$, *Selected- I^{Mh}* , *Action*)

Input: the update operation, $U^{Mh}(R)$ and the selected integrity constraints that might violate the $U^{Mh}(R)$, *Selected- I^{Mh}* with its associated integrity tests

Output: a call to the second level or action to perform the update $U^{Mh}(R)$

1. Begin
2. *SecondLevel* = { }
3. For each I^{Mh}_j in *Selected- I^{Mh}* do
4. Begin
5. Invoke the sufficient test, $ST-I^{Mh}_j$ of I^{Mh}_j if any
6. If there is no $ST-I^{Mh}_j$ or $ST-I^{Mh}_j$ is false then *SecondLevel* = *SecondLevel* \cup I^{Mh}_j
7. End
8. If *SecondLevel* \neq { } then
Action: Call *SecondLevel*(*MHid*, *BSId*, $U^{Mh}(R)$, *SecondLevel*, *Action*)
Else Action: Perform the $U^{Mh}(R)$
9. End

Procedure *SecondLevel*(*MHid*, *BSId*, $U^{Mh}(R)$, *SecondLevel*, *Action*)

Input: the mobile host *ID*, *MHid*, the base station *ID*, *BSId*, the update operation, $U^{Mh}(R)$ and the integrity constraints that need checking, *SecondLevel* with its associated integrity tests

Output: a call to the third level or a notification to perform the update $U^{Mh}(R)$

1. Begin
2. *ThirdLevel* = { }
3. For each I^{Mh}_j in *SecondLevel* do
4. Begin
5. Invoke the sufficient test, $ST-I^{Mh}_j$ of I^{Mh}_j if any
6. If there is no $ST-I^{Mh}_j$ or $ST-I^{Mh}_j$ is false then
7. Begin
8. Invoke the complete test, $CT-I^{Mh}_j$ of I^{Mh}_j
9. If $(CT-I^{Mh}_j(R) \cap R^{BC} = \{ })$, where *BC* is the current base station) or $(CT-I^{Mh}_j$ with existential quantifier is false) then *ThirdLevel* = *ThirdLevel* \cup I^{Mh}_j
Else If the $CT-I^{Mh}_j$ with universal quantifier is false then
10. Begin
11. Action: Notify the mobile host that violation occurs, *Notify*(*MHid*, *BSId*, *Abort*)
12. Exit()
13. End
14. Else If $CT-I^{Mh}_j$ with universal quantifier is true then


```

15. Begin
16.    $ThirdLevelI = ThirdLevelI \cup I^{Mh}_j$ 
17.   Action: Vote: perform the  $U^{Mh}(R)$ ,  $Notify(MHId, BSId, Vote: Perform)$ 
18. End
19. End
20. If  $ThirdLevelI \neq \{\}$  then Action: Call  $ThirdLevel(MHId, BSId, U^{Mh}(R), ThirdLevelI,$ 
     $Action)$  Else Action: Notify the mobile host to perform the  $U^{Mh}(R)$ ,
     $Notify(MHId, BSId, Perform)$ 
21. End
22. End
Procedure  $ThirdLevel(MHId, BSId, U^{Mh}(R), ThirdLevelI, Action)$ 
Input: the mobile host ID,  $MHId$ , the base station ID,  $BSId$ , the update operation,  $U^{Mh}(R)$ 
and the integrity constraints that need checking,  $ThirdLevelI$  with its associated integrity
tests
Output: a notification to abort or perform the update  $U^{Mh}(R)$ 
1. Begin
2. For each  $I^{Mh}_j$  in  $ThirdLevelI$  do
3.   Begin
4.   Invoke the sufficient test,  $ST-I^{Mh}_j$  of  $I^{Mh}_j$  if any
5.   If there is no  $ST-I^{Mh}_j$  or  $ST-I^{Mh}_j$  is false then
6.   Begin
7.   Invoke the complete test,  $CT-I^{Mh}_j$  of  $I^{Mh}_j$ 
8.   If  $CT-I^{Mh}_j$  with universal quantifier is false then
9.   Begin
10.  Action: Notify the mobile host that violation occurs,  $Notify(MHId, BSId, Abort)$ 
11.  Exit()
12.  End
13. End
14. End
15. Action: Vote: perform the  $U^{Mh}(R)$ ,  $Notify(MHId, BSId, Vote: Perform)$ 
16. End
Procedure  $Notify(MHId, BSId, Action)$ 
Input: the mobile host ID,  $MHId$ , the base station ID,  $BSId$ , the action,  $Action$ 
Output: execute the action,  $Action$ 
1. Begin
2. Read  $Action$ 
3. If  $Action = Abort$  then abort the update operation,  $U^{Mh}(R)$ , and other  $Action$  from other
   base station (if any) will be ignored, Exit()
4. If  $Action = Perform$  then perform the update operation,  $U^{Mh}(R)$  and other  $Action$  from
   other base station (if any) will be ignored, Exit()
5. If  $Action = Vote: Perform$  then analyze the votes received so far. The mobile host waits
   until it receives an action from one of the base station whose action falls under steps 3 or
   4 or majority of the base stations vote for  $Perform$ .
6. End

```

Fig. 3. The procedures of the Three-Level (3-L) model

Theorem 1. Given an update operation, $U^{Mh}(R)$, submitted at a mobile host, it is sufficient to check I^{Mh} , i.e. if I^{Mh} is satisfied then this implies that I^v is satisfied. (Note that this theorem does not state that the process of checking I^{Mh} is performed only at

h , it might involve the whole mobile system depending on the scope covered by the I^{Mh} , more specifically by the test selected which is associated to the I^{Mh} .)

Proof. For each $I_j \in I^0$, if $I_j(R) \cap R^{Mh} \neq \{\}$, then I_j is located at h , i.e. $I_j \in I^{Mh}$. Given an $I_k \in I^0$ and $I_k \notin I^{Mh}$, then $I_k(R) \cap R^{Mh} = \{\}$. Let I^{-Mh} denotes this set of constraints, thus $I^0 = I^{Mh} \cup I^{-Mh}$. Since the set of constraints I^{-Mh} does not contain the relation R in its specification therefore the I^{-Mh} is not violated with respect to $U^{Mh}(R)$. While the set of constraints I^{Mh} contains the relation R in its specification thus this set of constraints needs to be checked. Thus checking the I^{Mh} is sufficient.

Now let us consider a simple example to clarify the above model. Referring to the example given in Figure 1, assume that an update operation, $insert(emp(a, b, c, d))$, is submitted by the mobile host $M1$. Also, assume that there are only two base stations, $B1$, which is in the same cell as $M1$ and $B2$, which is the remote base station. Due to the limited capacity of the mobile host, only part of the emp relation is located at $M1$, while other relations are scattered between the base stations. Table 2 presents the possible flows in the Three-Level (3-L) model. Note that $I^0 = \{I_1, I_2, I_3, I_4\}$, $I^{Mh} = \{I_1, I_2, I_3\}$ and $Selected-I^{Mh} = \{I_1, I_3\}$. Based on the given example, we observe the Three-Level (3-L) model has the following benefits:

- The process of checking the constraints is performed at three different levels that span different sizes of areas. This reduces the amount of data accessed in particularly if the checking process involves only the first-level without having to go through the second and third levels.
- The model which supports both types of tests, namely: complete and sufficient, further reduces the complexity of checking the constraints. Adopting the sufficient test in this model is due to the characteristics of the test, which are (i) able to infer the information stored at the remote site(s), and (ii) give the opportunity to utilize as much as possible the information stored at the local site. These characteristics are suitable for a mobile environment in particular when the mobile hosts are disconnected from the entire system.

Table 2. Example flows of the Three-Level (3-L) Model

Level	Variable	I	Test	Action
1 $M1$	$Selected-I^{Mh} = \{I_1, I_3\}$ $SecondLevelI = \{I_3\}$	I_1	2	Assume that the $ST-I^{Mh}$, 2, is true, thus the I_1 is satisfied and further checking at the second level is not required.
		I_3	8	Assume that the $ST-I^{Mh}$, 8, is false. Thus, I_3 needs to be checked at the second level. (Otherwise, the I_3 is satisfied and checking at the second level is not required.)
2 $B1$	$SecondLevelI = \{I_3\}$ $ThirdLevelI = \{I_3\}$	I_3	8	Assume that the $ST-I^{Mh}$, 8, is false, then $CT-I^{Mh}$, 7, is checked. (Otherwise, the I_3 is satisfied and $M1$ can perform the update operation.)
			7	Assume that $CT-I^{Mh}_j(emp) \cap R^{B1} = \{\}$, thus the I_3 needs to be checked at the third level. (Otherwise, if the $CT-I^{Mh}_j$, 7, is not satisfied then violation is detected, else I_3 needs to be checked at the third level.)
3 $B2$	$ThirdLevelI = \{I_3\}$	I_3	8	Assume that the $ST-I^{Mh}$, 8, is false, then $CT-I^{Mh}$, 7, is checked. (Otherwise, the $CT-I^{Mh}$, 7, is omitted and $M1$ can perform the update operation.)
			7	If the $CT-I^{Mh}_j$, 7, is satisfied $M1$ can perform the update operation otherwise violation is detected.

5 Discussion of Performance

The key problem in integrity checking is how to efficiently evaluate the proposed checking mechanism. In this section, we estimate the performance of the Three-Level (3-L) model with respect to the following parameters [6]. We use the symbol $C(R_1, R_2, \dots, R_n)$ to denote the set of relations or fragment relations specified in the constraint or simplified form (test) C ; and MH , LBS and RBS to represent mobile host, local base station and remote base station, respectively.

- \mathcal{A}^L provides an estimate of the amount of data accessed, which is related to the number and the size of the relations or fragment relations specified in a given constraint or test, where L denotes MH , LBS or RBS . This measurement indirectly indicates the size of the checking space. It is based on the following formula: $\mathcal{A}_{C(R_1, R_2, \dots, R_n)} = \delta R_1 + \delta R_2 + \dots + \delta R_n$ where the R_i 's are the relations or fragment relations specified in C and δR_i is the size of R_i . To simplify δST^L , δCT^L , represent the amount of data accessed during the evaluation of the sufficient test and complete test at L , respectively. $[\mathcal{A}_{min}, \mathcal{A}_{max}]$ is a range where \mathcal{A}_{min} (\mathcal{A}_{max} , respectively) is the minimum (maximum, respectively) amount of data that might be accessed.
- σ gives a rough measurement of the size of area that might be involved in validating the constraint.

Figure 4 presents the possible flows during the evaluation of I_1 . Similar flows are observed for the other integrity constraints. Some conclusions can be made as follows:

- The first level which spans only the mobile host, i.e. $\sigma = MH$, accessed small amount of data, $\mathcal{A}^{MH} = \delta ST^{MH}$, where $\delta ST^{MH} < \delta R$. This is due to the characteristics of the sufficient test, which only accesses the data from the target relation (relation which appears in the specification of the update operation) and $\delta ST^{MH} < \delta R$ since only part of the relation R is stored at the mobile host due to its limited capacity. Therefore, at this level it is important to have a high rate of success. This can be achieved by caching the relevant data items that are required by each of the test to the mobile host during relocation period.
- The second level spans the local base station of the mobile host. Since this level is embarked once the first level failed to validate the constraints, therefore, $\sigma = MH + LBS$ (the operator $+$ denotes that the size of area covered by the checking process include both the mobile host and the local base station). The amount of data accessed, $\mathcal{A}^{LBS} = [\mathcal{A}^{MH} + \delta ST^{LBS}, \mathcal{A}^{MH} + \delta ST^{LBS} + \delta CT^{LBS}]$, i.e. $\mathcal{A}^{MH} + \delta ST^{LBS}$ if the sufficient test can verify if the initial constraint is not being violated. The worst case if when complete test needs to be evaluated and thus the amount of data accessed up to this level is $\mathcal{A}^{MH} + \delta ST^{LBS} + \delta CT^{LBS}$. Therefore, at this level, it is important to have a high rate of success of the sufficient test or the complete test.
- The third level spans the remote base station, which can involve more than one remote base station. Since this level is embarked once the second level failed to validate the constraints, therefore, $\sigma = MH + LBS + RBS$. The amount of data accessed, $\mathcal{A}^{RBS} = [\mathcal{A}^{LBS} + \delta ST^{RBS}, \mathcal{A}^{LBS} + \delta ST^{RBS} + \delta CT^{RBS}]$, i.e. $\mathcal{A}^{LBS} + \delta ST^{RBS}$ if the sufficient test can verify if the initial constraint is not being violated. The worst case if when

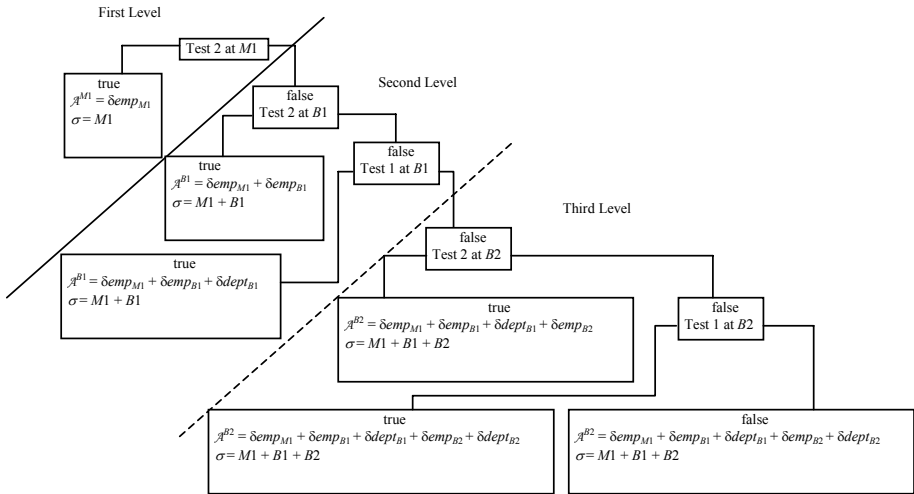


Fig. 4. The evaluation of I_1

complete test needs to be evaluated and thus the amount of data accessed up to this level is $\mathcal{A}^{LBS} + \mathcal{S}T^{RBS} + \mathcal{C}T^{RBS}$. This level is similar to the brute force strategy, which spans the entire mobile system. Nevertheless, it is seldom the case that the entire mobile system is enforced to validate the constraints. As one can notice the scenario represented at this level is the same scenario as appeared in the distributed databases and the parameters presented above can be significantly reduced by applying the same optimization strategies as used in distributed databases [6].

6 Conclusion

This paper has presented the Three-Level (3-L) model, which is designed for checking database integrity in a mobile environment. The model has three levels and the process of checking the constraints embarks on at the first level. In this level, the information that is stored at the mobile host is accessed in order to check for the constraint violations. The second level which checks the validity of the constraints is performed at the base station of the mobile host by accessing the information stored at the station. The third level is invoked only if the second level fails to guarantee the constraint violations. The third level accesses the information stored at the remote base station(s). This model which adopts the simplified forms of integrity constraints, namely: sufficient and complete tests, together with the idea of caching the relevant data items during the relocation period for the purpose of checking the integrity constraints has reduced the amount of data accessed given that much of the tasks are performed at the mobile host. Eventually the checking mechanism of mobile databases is improved as delay during the process of checking the integrity constraints and performing the update is reduced. For future work, we plan to measure the performance of the Three-Level (3-L) model with respect to the time taken in checking the integrity of the mobile databases.

References

1. Alwan, A.A., Ibrahim, H., Udzir, N.I.: Local Integrity Checking using Local Information in a Distributed Database. In: Proceedings of the 1st Aalborg University IEEE Student Paper Contest 2007 (AISPC 2007), Aalborg (2007)
2. Epfl, Grenoble, U., Inria-Nancy, Int-Evry, Montpellier, U., Paris, U., Versailles, U.: Mobile Databases: a Selection of Open Issues and Research Directions. SIGMOD Record 33, 78–83 (2004)
3. Gupta, A.: Partial Information Based Integrity Constraint Checking. PhD Thesis, Stanford University, USA (1994)
4. Hanandeh, F.A.H.: Integrity Constraints Maintenance for Parallel Databases. PhD Thesis, UPM, Malaysia (2006)
5. Ibrahim, H.: Checking Integrity Constraints – How it Differs in Centralized, Distributed and Parallel Databases. In: Proceedings of the Second International Workshop on Logical Aspects and Applications of Integrity Constraints (LAAIC 2006), Krakow, pp. 563–568 (2006)
6. Ibrahim, H., Gray, W.A., Fiddian, N.J.: Optimizing Fragment Constraints – A Performance Evaluation. International Journal of Intelligent Systems – Verification and Validation Issues in Databases, Knowledge-Based Systems, and Ontologies 16(3), 285–306 (2001)
7. Ken, C.K.L., Wang-Chien, L., Sanjay, M.: Pervasive Data Access in Wireless and Mobile Computing Environments. Journal of Wireless Communications and Mobile Computing (2006)
8. Madiraju, P., Sunderraman, R.: A Mobile Agent Approach for Global Database Constraint Checking. In: Proceedings of the ACM Symposium on Applied Computing (SAC 2004), Nicosia, pp. 679–683 (2004)
9. Martinenghi, D.: Advanced Techniques for Efficient Data Integrity Checking. PhD Thesis, Roskilde University (2005)
10. Mazumdar, S., Chrysanthis, P.K.: Localization of Integrity Constraints in Mobile Databases and Specification in PRO-MOTION. In: Proceedings of the Mobile Networks and Applications, pp. 481–490 (2004)
11. McCarroll, N.F.: Semantic Integrity Enforcement in Parallel Database Machines. PhD Thesis, University of Sheffield, UK (1995)

Usability Issues of e-Learning Systems: Case-Study for Moodle Learning Management System

Miroslav Minović, Velimir Štavljanin, Miloš Milovanović, and Dušan Starčević

Belgrade University, Faculty of Organizational Sciences, Jove Ilica 154, Belgrade, Serbia
{mminovic, velimirs, milovanovicm, starcev}@fon.bg.ac.yu

Abstract. Mobile devices have potential to be integrated into the classroom, because they contain unique characteristics such as: portability, social interactivity, context sensitivity, connectivity and individuality. Adoption of LMS by students is still on the low rate, mostly because of poor usability of existing eLearning systems. Usability issue is rising to the higher level on mobile platform, due to device limitations and also because of context of use. Our hypothesis was that it is wrong to take a mobile device as a surrogate for desktop or laptop PC. By accessing LMS on mobile devices using adaptive technologies, like Google proxy, we didn't acquire the satisfactory results. Possible solution to the problem could be development of rich client applications for today mobile devices that would improve usability. Results gathered in usability research conducted among students have confirmed that development of eLearning systems needs to have learner in the center of development process.

Keywords: Usability, User center design, mLearning, Moodle, Mobile devices.

1 Introduction

Education is organized process of knowledge, skills, values and beliefs transfer and prerequisite for any improvement at individual or social level. Due to technological advances new opportunities emerge to fulfill the process of education amongst the strongest representative is the computer, which with its abilities added a whole new dimension to the education process [1]. E-learning is an approach to facilitate and enhance learning through both computer and communications technology. This type of learning uses network that can be Internet, university network or corporate computer network.

E-learning is usually based on learning management systems LMS. LMS is software for different types of direct and indirect interaction between professors and students, and exchange of different type of electronic learning material. Most used systems are Blackboard, WebCT (commercial software) and Moodle (free open source software).

In order to truly integrate eLearning system into regular curriculum at University, mobile access to LMS has to be enabled. Mobile devices have potential to be integrated into the classroom, because they contain unique characteristics such as: portability, social interactivity, context sensitivity, connectivity and individuality. But

student experience is not always good, and adoption of LMS by students is still on the low rate. This is mostly because of poor usability.

The prime assumption of this work is that poor usability of existing eLearning systems leads to poor adoption. Our second hypothesis is that it is wrong to take a mobile device as a surrogate for desktop or laptop PC. By just adopting existing LMS on mobile devices with adaptive technologies, like Google proxy, we do not acquire the satisfactory results. Usability can prove to be even lower compared to desktop application.

This paper is aimed at issues of LMS systems usability for desktop platform as well as mobile devices. Those issues are addressed to in section two of this paper. Existing research in this field is a focus of section three. As a competitive technology for our usability study we developed a prototype that we presented in section four. Above mentioned usability study as well as results are presented and discussed in section five. Conclusion is given at the end of the paper.

2 Usability Issue of e-Learning Systems

As a part of our teaching activities, our faculty is using Moodle LMS (Learning Management System) in order to support course activities. Professors are usually adding contents for a course, on a weekly basis. Students are provided with the ability to regularly inform on new events and gain new information on course via News section on our eLearning portal. Collaboration, as well as discussion is encouraged through forums. Quiz module is widely used for student self-examination during the semester, and also for student knowledge evaluation. In spite of the obvious upsides of this type of conducting course, students brought to our attention several issues of use. We are constantly receiving e-mails, with questions about finding some material, logging-in to the system or grade checking. Students often get frustrated with these problems which are providing the reasons for complaint.

On the other side, same problems occurred during our collaboration with Energo-projekt company that resulted in building a life-long eLearning system through utilization of Moodle LMS for knowledge verification [2, 3].

Several questions are raised from this experience: Is Moodle too complicated for novice users? Is there a usability problem with Moodle?

Also we cannot disregard the learning effect that can be achieved “*On the go*”. Standard use of LMS systems simply by use of desktop computer does not fully involve the user and it cannot provide essential information at any time. One solution to that problem is provided by mobile technologies. By using adaptive technologies we can reformat the content to suite mobile devices. The problem is that by doing so we usually end up with confusing content due to limitations of such device.

This lead us to our research hypothesis: Moodle LMS has usability issues, which represents major disadvantage of this LMS, and makes positive aspects of eLearning systems less effective; Usability issues are rising to the higher level on mobile platform; It is wrong to take a mobile device as a surrogate for desktop or laptop PC.

3 Existing Research in the Field

Multimodal interaction is part of everyday human discourse: We speak, move, gesture, and shift our gaze in an effective flow of communication [4]. While multimodal interaction research focuses on adding more natural human communication channels into HCI, accessibility research is looking for substitute ways of communication when some of these channels, due to various restrictions, are of limited bandwidth [5]. During our research we addressed general issues of multimodal HCI and universal accessibility by proposing generic frameworks [4, 5]. A specific area of our research is dedicated to usability issues of e-Learning systems and mobile devices.

There is a huge bibliography on adaptive and context aware applications [6]. In particular, lots of papers that have been written on this issue in the context of mobile computing: adaptation to limited device capabilities, network bandwidth, location, QoS and user preferences (among others) have been already deeply studied. However, research area targeting access to Moodle via mobile devices is not adequately addressed to, only few solutions for mobile access to Moodle content was proposed [7,8,9,10], and few researches were conducted concerning usability of Moodle via mobile devices.[11,12] On the other hand, the usability issues for mobile devices were a common subject among many researches, which shows the effectiveness of experimental method applied in our usability research [13]. Also there were several projects not specifically targeting Moodle, but offering solutions for social interaction via mobile devices [14, 15, 16, 17] as well as custom made m-learning solutions [18,19,20,21].

Most of the researches rely on adaptive technologies in providing access to eLearning systems e.g. Mobile browsers. However, this approach has few drawbacks:

Limited screen size: Standard Moodle pages are designed for access from standard desktop PC, with large screens, but mobile devices have very limited screen size. Mobile browser wrap content in order to show it in whole, and we lose initial page layout. Browsing through standard web pages by use of mobile browsers is at a low level.

Limited input methods: Input on standard web pages strongly relies on keyboard usage, but mobile devices usually lack one.

Limited network bandwidth: Each web page is actually bunch of HTML code, and each page load and reload actually sends request to the web server and receives the whole HTML code for requested page. Network overhead can be pretty big, when we open standard web pages via mobile device.

4 Rich Client Prototype

In order to test our hypothesis we decided on comparing standard approach to Moodle LMS via desktop computers against mobile solutions. Since usability of LMS systems is subject of test, we also required a comparison technology for adaptive mobile solution. For that purpose we decided to develop a rich client application for PocketPC, and a Web service as standard middleware interface between Moodle database and a client application. System architecture is shown below (Fig. 1).

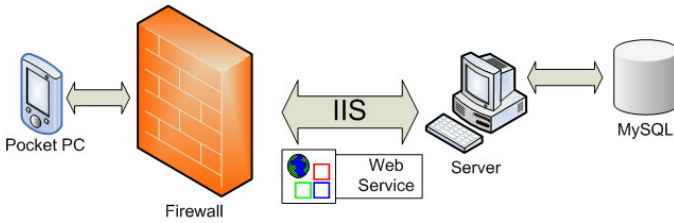


Fig. 1. Mobile Moodle architecture

Since Moodle was developed using PHP/MySQL platform, we have chosen to develop a Web Service as more universal data source to access Moodle from different kinds of devices and platforms. Because Web Service implements standard interface described by WSDL, accessible via SOAP based on XML it's very well suited as a universal data source, much better than just MySQL database. It also supports additional features such as using a firewall for extensive security without additional reconfiguring. In our architecture, Web Service is very important, in order to develop clients and support broad range of mobile devices (PDAs, mobile phones, smart phones, etc.).

Rich client application could be a better solution than standard or WAP Moodle pages, because it targets main drawbacks (listed in previous section) of these solutions. However there are downsides to using rich client (fat client) where most significant is *forking* (Certain changes to Moodle will require updates of client as well as server).



Fig. 2. News module (list and detail) and activity module (list of activities between chosen dates)

Brief comparison between existing solutions and prototype application by selected criteria is given in the table below.

Table 1. Comparison of HTML,WAP and rich client Moodle access

	HTML/WML Moodle	Rich client Moodle
Limited screen size	HTML controls, one page model – harder to use	Better component layout – easier to use
Limited input methods	each user action require response from the server, over the network – slower and less productive	Richer user controls gives more options for user interaction – faster and more productive
Limited network bandwidth	has network overhead, complete page is reloaded for each data change	small network overhead, only new/modified data exchanged

Brief comparison showed competitive advantages of customized smart client application over HTML/WML based solution. Based on that we proceeded with usability study, which includes examination of different usability aspects such as: stability, response and feedback, consistency, control and screen design.

5 Usability

Usability often refers as the question of how well users can use system functionality [22]. Usability is not one-dimensional property of user interface. It's associated with five attributes: learnability, efficiency, memorability, errors and satisfaction. In order to measure usability we conducted a Think aloud study [22] amongst University students.

The goal of the study was to determine the usability of Moodle LMS system. We attempted to determine the quality of our PDA application prototype in comparison to other available technologies for using Moodle via mobile devices and also to compare the results to standard desktop approach using web browser. As the alternative technology we have chosen the *Google Proxy* for mobile devices that provides the service of reformatting the requested content to be more suitable for mobile devices. We used *Google Proxy* for mobile phone and PDA as well.

Our research was conducted on one Desktop PC, two PDA devices and two mobile phones. First PDA was *HP Ipaq rx3715*, with our rich client prototype. Second was *Dell Axim X30*, that subjects used for performing tasks on *Google Proxy* reformatted content. Both devices had *Pocket PC 2003* for an operating system. Finally, our mobile phone devices were *Nokia*, model *N80* with Symbian OS 9.1 and slider numerical keyboard, as well as *QTEK*, model *9100*, with Windows Mobile 5, touch screen and slider QWERTY keyboard.

Students first performed a predefined set of tasks on a desktop computer using web browser. Then they performed the same predefined set of tasks firstly on PDA using our custom PDA Application, following on PDA using internet browser through *Google Proxy*, and at the end on mobile device using internet browser through *Google Proxy*. The tasks were done in a predefined order. First they had to log in. Then they were expected to check for news and then read them. Next came checking for the

upcoming activities and informing on them. Following, they needed to send a message to other participant as well as check for their own messages. Finally they were required to check their grades on different courses.

After the participants performed a set of tasks on different platforms, they were asked to fill out a questionnaire. Questionnaire included a few demographic questions about respondents and their computer skills. Then followed questions about subjective satisfaction on every platform and questions that required them to rate the platforms and to explain their rating. Questions about subjective satisfaction were presented using seven points semantic differential rating scale from positive impression to negative impression (for example 1 = complicated 7= simple).

Subjects in our research were undergraduate senior year students from different departments at University of Belgrade Faculty of organizational sciences. Research was conducted in a laboratory conditions. A total of 12 students participated in a study and all of them completed the end survey. Respondents were 8 men and 4 women. All respondents were experienced users of computer, PDA and mobile phone. The mean knowledge about CMS systems was 4.92, on the seven point scale, where 1 = no knowledge about CMS systems, and 7 = sufficient knowledge about CMS systems. On the scale ranging from 1 = little experience with e-learning to 7 = experienced user of e-learning systems, our participants mean was 4.58, with no answer under 3.

Students performed the tasks while sitting down. They were documented by two cameras, one aimed directly at their face to reveal facial expressions during the session and another aimed covering actions on the mobile device. Also a microphone placed on the subject recorded commentary and voice. During the session the subjects were encouraged to think out loud, by asking them questions such as: „*What are your thoughts now?*“, and „*Can you state your impressions about performing this action?*“.

During the task completion we measured efficiency of use by measuring number of clicks/taps and the times necessary to complete the task. Besides efficiency we measured errors by number and type (simple and catastrophic), and subjective satisfaction.

First table (Table 2.) provides the results of measuring the amount of click/tap actions to complete the given operation with results of measured amount of data transfer in Kb per operation. Operations are processed for each device/technology. The results provided indicate that PDA Application has the lowest amount of click/tap actions comparing to other technologies. The only exception is *Read Activities*. The reason for that is poorly developed input control for specifying the date interval for searching the activities. It does not provide the ability of choosing the date from calendar but requires manual input. Another indicative that this is a good place of improving the interface came from our test subject that commented on this feature as inadequate during our *Think aloud* study. Some of these comments were: “*The date input is too complicated!*” or “*It is too difficult to enter the date, and I am repeatedly making a mistake!*”.

The given data for data transfer clearly states the obvious advantage for PDA Application comparing to other technologies. Interaction between PDA Application and a Web Service provides impressive amount of savings in data transfer due to the ability to return only the data relevant for the given operation.

Table 2. Click or Tap numbers/ Measured data transfer (Kb)

	Desktop		PDA Application		PDA Browser		Mobile Browser	
	No	Kb	No	Kb	No	Kb	No	Kb
Login	15	166	15	1	16	37	24	37
Read News	2	17	1	4	7	13	12	13
Read Activities	2	30	22	1.5	9	16	12	16
Send Message	17	7	16	1	22	9	29	9
Receive Message	2	6	1	2	5	7	8	7
Check Grades	3	5	1	1.5	6	4	9	4

Second table (Table 3.) is a summary of results acquired by measuring time efficiency of each operation executed by our test subjects. The data shown in table are average times per operation for given devices/technologies. Revision of data leads us to a conclusion that PDA Application is more time efficient than other two mobile technologies for each operation performed. Interesting fact is that it also proven to be more efficient than standard Desktop use of Moodle except in two cases *Login* and *Read Activities*. Average time for *Read Activities* can be explained by poor method of date input mentioned earlier while the reason of longer lasting *Login* operation could be blamed on lack of keyboard on PDAs part. Also several of our test subjects positively commented on ease of use of PDA Application as opposed of Desktop internet browser. Some of these comments were: “*It is a bit confusing to navigate to the wanted section, and it is hard to immediately find a way to perform the given operation*”, this regarding the Desktop internet browser, and also “*It is much simpler to find my way around on this than on Desktop*”, regarding the PDA Application. The results and subject comments lead us to a conclusion that Moodle is not intuitive and user friendly. It is obvious that our subjects had difficulty in performing even the easiest of tasks using this technology.

Table 3. Average user time per operation (second), for each of devices

Time (second)	Desktop	PDA Application	PDA Browser	Mobile Browser
Login	27.8	34.7	39	54.3
Read News	58.2	23.2	80.6	87.4
Read Activities	82.6	98.5	121.5	139.9
Send Message	74.8	39	181.6	209
Receive Message	55	27	65.9	57.2
Check Grades	45	18.8	59.6	65.6

In order to graphically present the corresponding data we provided the chart (Fig. 3). Average time per operation, for our rich client prototype is shown with red vertical bar.

As we described subjective satisfaction was measured by seven point’s semantic differential rating scale. Questions included in measurement were: System is pleasant to use; Interface is complete; Interface is simple for use; System is fast for use; System is cooperative in completing the tasks. Results are shown on the chart (Fig. 4).

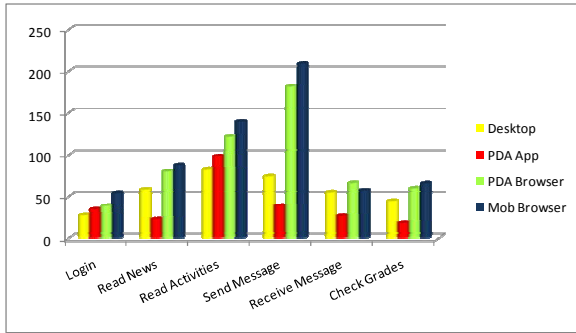


Fig. 3. Average user time per operation (seconds), for each of devices

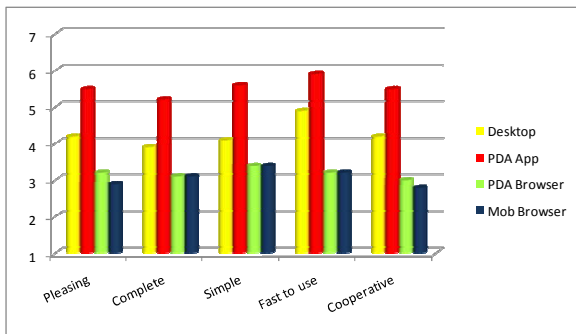


Fig. 4. Results describing subjective satisfaction for each platform

Rankin results were similar as results from satisfaction measurement. Almost all respondents (9 of them) said that the most preferred platform is PDA application. Second preferred was desktop, third PDA browser and fourth Mobile browser. Some comments about PDA platform were “PDA application is very easy for use, and almost all poor implementations from desktop are corrected.” or “PDA application is almost perfect!” or “Definitely, PDA application is my the most preferred solution”.

During the test there were no catastrophic errors, but there were few occurrences of simple errors such as accidental closing of mobile browser (two times) and one network error during the call to a Web service. The test resumed after the second try.

In spite of the positive results acquired by this research we noticed a few downsides to this type of testing. Primarily, the order of conduct implied the ability of our test subjects to accommodate to the LMS’s way of use. Since they performed the same set of tasks by use of Desktop, PDA Application, PDA browser and mobile browser respectively they were in position to learn how to perform the same tasks on mobile devices with adaptive technologies. Even so, the results clearly stated that PDA browser and mobile browser were by far the most complicated tools in order to complete the given tasks. Limited resources provided us with another difficulty during our session. The lack of instruments forced us to form a queue, which caused the

need of additionally motivating our subjects. This is also a reason why the optimal amount of test subject was only 12.

Due to mobility of technology tested here, we cannot ignore the effect of using eLearning system “*On the go*” which is probably the strongest argument for this type of technology. Next step in our research will be to conduct a study in real life situation, away from office or classroom, and to consider the usability in such circumstances. Also we should consider the learning effect achieved this way.

6 Conclusion

During our experience in working with LMS’s we came to a conclusion that users have a problem accommodating to them. Another question that occurred was inability of such systems to adequately provide their services via mobile devices. For that purpose we conducted a usability study that targeted user’s ability to accommodate to specific LMS. As an alternative to mobile adaptive technologies for access to specific LMS we developed a rich client prototype for mobile device. Our usability study included this technology as an alternative.

The results and subject comments gathered during our study lead us to a conclusion that Moodle is not intuitive and user friendly. It stated as obvious that our subjects had difficulty in performing even the easiest of tasks using desktop technology. Adaptive technologies for accessing Moodle via mobile devices gave even lower results, and proved as inadequate. Our rich client prototype proved as more time efficient than other two mobile technologies for each operation performed. Interesting fact is that our prototype even preceded desktop approach and was favored by most of the subjects. Further development may include implementation of other popular Moodle modules (like blog, wikis, quiz, hot potatoes quiz, lessons, assignments...). However, we should carefully weight benefits before deciding to implement support for other Moodle modules in rich client application, because of mobile device limitations (e.g. screen size, memory, keyboard). Not all of them are well suited to be used from mobile device.

As a continuation of our research we will focus on usability of LMS systems in real life situation, during the class and also away from office or classroom, by use of mobile devices.

Acknowledgements. This work is part of a project “Corporate portal for employee long life learning”, funded by the Ministry of science and technology Republic of Serbia, grant no: 006221.

References

1. Wikipedia, <http://www.wikipedia.org>
2. Pantovic, V., Starcevic, D., Savkovic, M.: Virtual Business School of Energoprojekt Group. In: Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2006, pp. 833–838. AACE, Chesapeake (2006)
3. Pantovic, V., Starcevic, D., Savkovic, M.: The Role Of Portal Technologies In Corporate Lifelong Learning System. In: Proc. of the CATE 2006, Lima, Peru (2006)

4. Obrenovic, Z., Starcevic, D.: Modeling multimodal Human-Computer interaction. *IEEE Computer* 37(9), 62–69 (2004)
5. Obrenovic, Z., Abascal, J., Starcevic, D.: Universal accessibility as a multimodal design issue. *Communications of the ACM* 50(5), 83–88 (2007)
6. Corradi, A., Montanari, R., Toninelli, A.: Adaptive Semantic Middleware for Mobile Environments. *Journal of Networks* 2(1) (2007)
7. Houser, C., Kinjo, T.P.: Poodle: a course-management system for mobile phones. In: *IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE 2005)*, pp. 211–215 (2005)
8. Sharples, M., Corlett, D., Westmancott, O.: The Design and Implementation of a Mobile Learning Resource. *Personal and Ubiquitous Computing* 6(3), 220–234 (2002)
9. Yingling, M.: Mobile Moodle. *Journal of Computing Sciences in Colleges* 21(6), 280–281 (2006)
10. Bar, H., Haussge, G., Rosling, G.: An Integrated System for Interaction Support in Lectures. In: *ITiCSE 2007, Dundee, Scotland, United Kingdom* (2007)
11. Kramer, B.J., Strohlein, G.: Exploring the Use of Cellular Phones for Pervasive eLearning. In: *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW 2006)*, pp. 190–195 (2006)
12. Seong, D.S.K.: Usability Guidelines for Designing Mobile Learning Portals. In: *The 3rd International Conference on Mobile Technology, Applications and Systems - Mobility 2006, Bangkok, Thailand* (2006)
13. Bodén, J., Jegers, K., Lidström, M., Wiberg, C., Wiberg, M.: Point or click? In: *Second International Conference on Internet and Web Applications and Services ICIW 2007* (2007)
14. Chatti, M.A., Srirama, S., Kensch, D., Cao, Y.: Mobile Web Services for Collaborative Learning. In: *Fourth IEEE International Workshop on Wireless, Mobile and Ubiquitous Technology in Education ICHIT 2006* (2006)
15. Counts, S., Hofte, H.T., Smith, I.: Mobile Social Software: Realizing Potential, Managing Risks. In: *CHI 2006, Montréal, Québec, Canada, April 22–27* (2006)
16. Beale, R.: Mobile blogging: supporting informal mobile learning. In: *MLEARN 2005, Cape Town, South Africa* (2005)
17. Beale, R.: Supporting Social Interaction with Smart Phones, *PERVASIVEcomputing* (April–June 2005)
18. Black, J.T., Hawkes, L.W.: A Prototype Interface for Collaborative Mobile Learning. In: *IWCMC 2006, Vancouver, British Columbia, Canada* (2006)
19. Sharples, M., Corlett, D., Westmancott, O.: The Design and Implementation of a Mobile Learning Resource. *Personal and Ubiquitous Computing* 6(3), 220–234 (2002)
20. Zhang, Y., Zhang, S., Vuong, S., Malik, K.: Mobile Learning with Bluetooth-based E-learning System. In: *IWCMC 2006, Vancouver, British Columbia, Canada* (2006)
21. Costabile, M.F., De Angeli, A., Lanzilotti, R., Ardito, C., Buono, P., Pederson, T.: Explore! Possibilities and Challenges of Mobile Learning. In: *CHI 2008, Florence, Italy* (2008)
22. Nielsen, J.: *Usability Engineering*. Morgan Kaufmann, San Francisco (1993)

CPL: Enhancing Mobile Phone Functionality by Call Predicted List

Santi Phithakkitnukoon and Ram Dantu

Dept. of Comp. Sci. & Eng., University of North Texas, Denton, TX 76203, USA
{santi, rdantu}@unt.edu

Abstract. In this paper, we present a concept of a new advanced feature for a mobile phone that provides its user functionality for predicting future calls. The feature is envisaged as a Call Predicted List (CPL) which makes use of the user's call history to build a probabilistic model of calling behavior based on the caller's calling patterns and reciprocity. The calling behavior model is then used to generate a list of numbers/contacts that are the most likely to be callers in the next hour. The performance of the CPL is evaluated with the real-life call logs and it shows promising results in accuracy.

Keywords: Context-aware computing, Call prediction, Mobile phone.

1 Introduction

With the rapid development of telecommunication technologies and the fast-growing number of users on the networks, the mobile phone has moved beyond being a mere technological object and has become an integral part of many people's lives. The mobile phone is gradually becoming the ubiquitous computing device at this early stage of the pervasive-computing era where handheld devices are precursors to a phase of ambient computing that is always on, personalized, context-sensitive, and highly interactive.

Mobile phones record the history of our lives in the form of the call logs. Utilizing these call logs in computing human (user)'s behaviors can indeed enhance the capability of the mobile phone as it is becoming more than just a communication device but also an intelligent assistant to its user.

In this paper, we present a novel model for predicting future callers using calling patterns. In this way, the mobile phone becomes more responsive and sensitive to the user's context and needs. With our proposed model, the personal phone will become more intelligent as it learns the user's behavior over time as well as the behavior of those who call the user in order to provide the most accurate prediction possible of the future incoming caller for the user upon his/her request. The rest of this paper is structured as follows: Section 2 presents the concept of the Call Predicted List (CPL), Section 3 presents the CPL's framework which describes the behavior learning model, Section 4 discusses the performance of the CPL, and the paper is concluded in Section 5 with a summary and an outlook on future work.

2 Call Predicted List

The Call Predicted List (CPL), described here, is intended to provide a phone user with an ability to predict future incoming calls as well as an improvement over the “last received calls” functionality that is often provided on today’s phones and communication clients (e.g. VoIP soft phones).

Quite often in our daily lives, we find ourselves in a situation where we wish to know who will be calling in the next hour so we could schedule (plan) things out accordingly. In many occasions that we know for certain that we will be unavailable to accept any incoming calls over the next hour (e.g. having a flight, attending a class, having a meeting) thus we wish to know who will be calling during the next hour so we could perhaps make a call to the persons to inform of our next-hour schedule as we do not wish to miss any important future calls which could be too important calls to miss.

The user interface on a today’s mobile phone normally provides easy access to a list of recently received numbers (contacts). The list provided in this case is insensitive to the user’s context. It only shows the most recently received numbers and therefore takes no account of other call related information (e.g. time, day of week, frequency, etc) to provide a better guess of the numbers that the user will find most useful.

Our CPL makes use of the user’s call history, i.e. call numbers received, time of call received, day of call received, frequency of call, and last dialed numbers, to build a probabilistic model of calling behavior. The calling behavior model is then used to generate a list of numbers/contacts that are the most likely to be the callers for the next hour. The list can be presented to the user in a number of different ways for different purposes. We envisage the CPL as an “intelligent call predicted list,” i.e. a list that anticipates the numbers/contacts that the user will receive in the next hour and gives these numbers (potential callers) higher precedence on the list. Figure 1(a) shows an example of the CPL where the most likely callers are listed higher on the list.

3 Call Predicted List Framework

In our daily life, when we receive a phone call, at the moment of the first phone ring and right before looking at the caller ID, we often guess who the caller might be. We often base this estimation on the caller’s calling pattern and our past communications to the caller.

Each caller tends to have a unique calling pattern. This pattern can be observed through history of *time of calls*, i.e. we normally expect a call from someone who has history of making several calls at some particular time of day. For example, your spouse likes to call you while you drive to work in the morning therefore when your phone rings while you are on the way to work you are likely to guess that it is a phone call from your spouse. We also base our estimation on day of calls, for example, your close friend has made several calls to you on every Tuesday because it is his day off

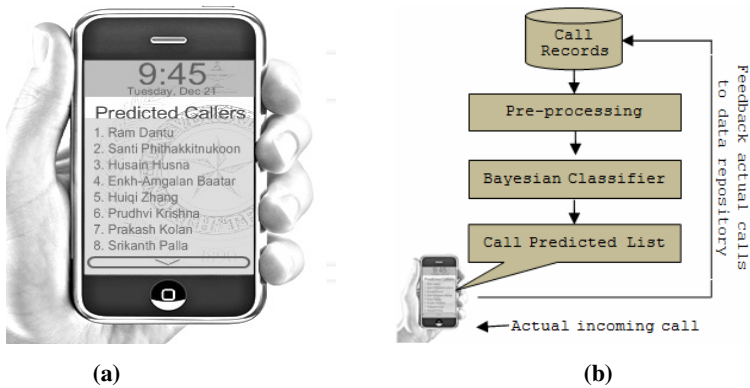


Fig. 1. (a) CPL user interface, (b) Basic system overview

therefore when your phone rings on Tuesday, the first person that comes to mind is your close friend. Similarly, the person who has made the most calls (*total call count*) to you (regardless of time and day) among other callers is also the person whom you most anticipate the calls from. Receiving a call is also influenced by the *reciprocity* or call interaction between user and caller. For example, you may expect a call from your friend based on your last phone conversation with him/her (e.g. “call me when you get home” or “call me same time tomorrow” or “I’m busy right now, call me back in an hour”). This reciprocity may sequentially lead to later receiving calls from that caller caused by your initiative. For example, you make a call to a friend to whom you have not called for a long time, and then you later receive calls back from this friend. Another example, you make a call to your mother to get some advice during the night (assume that you do not normally make or receive calls from her during this time), and then you receive calls from your mother later on during that night.

These are examples that actually happen in our everyday life for most of us who are phone users. Understanding the actual human behavior towards phone usage gives CPL an intelligence to assist its user effectively.

To predict the future incoming calls, the behavior learning model must be used. This model should incorporate mechanism for capturing caller’s calling behavior. Calling behavior of the caller can be observed via the call logs which can be obtained from a variety of sources. For example, they may be collected by a network or service operator for billing purposes or they may be captured directly on device such as a mobile phone or on a software application such as a VoIP softphone.

In our current implementation, we use a set of actual call logs collected from 20 mobile phone users at the University of North Texas. These 20 individuals are faculty, staff, and students. We are in process of collecting several more call logs and make them publicly available for other researchers who have interests. This call logs collecting process is a continuation of the Nuisance Project [2], where Kolan et al. studied the nuisance level associated with each phone call. The details of the data collecting process are given in [3].

As part of the data collecting process, each user downloaded three months of detail telephone call records from his/her online accounts on the mobile phone service provider's website. Each call record in the dataset had 5-tuple information as follows.

- Date – data of call
- Start time – start time of call
- Type – type of call, i.e. “Incoming” or “Outgoing”
- Call ID – caller/callee identification
- Talk Time – duration of call (in minutes)

The call record is subject to pre-processing to extract features or information about *time of calls* (day and hour), *total call count*, and *reciprocity*.

The pre-processed call records are eventually fed into the classifier to be ingested. Classifier then outputs a list of phone numbers ordered by the predicted likelihood of the number being the next-hour caller given time of calls, day of calls, total call count, and reciprocity. The basic system overview is shown in Fig. 1(b).

Classifier has two modes of operation; training and predicting. During the training, classifier ingests the pre-processed call logs and constructs four hash tables which primarily contain call counts and corresponding callers. The first table maps each unique telephone number (or caller identifier) to a count of calls received for each day of the week. The second table maps each unique telephone number to a count of calls received for each hour of the week. The third table maps each unique telephone number to the total number of calls received.

It is not trivial to quantify the *reciprocity*. Having no knowledge about the context of the phone calls from the user to the callers, it is difficult to identify which outgoing calls may influence future incoming calls. Nevertheless a received call can be linked to user's calling behavior which is recorded in the “last dialed calls” list (normally a list of last 20 outgoing calls) where the lower order corresponds to more recent dialed number (e.g. “1” is the most recent dialed number, “20” is the least recent dialed number). Thus the same number/contact can occupy in more than one position on the list. Clearly the numbers/contacts on the list are pushed down one position when new call is received.

Based on the position on the list and its corresponding number of times that actual incoming caller was listed on that position, the likelihood of receiving a call can be estimated. For example, suppose currently statistic (hash table) shows that position “3” of the list has the most counts, it implies that the number/contact that is on position “3” of the current “last dialed calls” list has the highest likelihood of being the next caller. Therefore the fourth hash table maps each position on the “last dialed calls” list to a count of calls received.

Once the input call records have been ingested and the hash tables generated, the classifier is considered trained. With the classifier trained on a set of representative call records, it is then ready to be used in predicting mode. The classifier is given a target day of week, hour of day, total call count, and current “last 20 dialed calls” list, and uses the calling behavior model to estimate the likelihood of the user receiving each of the telephone numbers (or caller identifiers) seen in the training data. Clearly the classifier can only make predictions for numbers that it has already seen.

A likelihood metric is calculated for each number known to the classifier and the numbers are then sorted in descending order of likelihood of being received. If the

caller's behavior has a degree of temporal predictability (i.e. they tend to make calls to user at a certain time of the day, or in a particular day of the week, or after some number of calls from the user), then it is expected that the number is likely to be listed towards the top of the list. When several numbers end up with the same value of likelihood, they are listed in alphanumeric order.

The classifier itself is of a type known as a Naïve Bayesian Classifier. In our case, we wish to compute the likelihood of each number (T_n) being received given that the day of the week (D_x), hour of the day (H_y), the current "last 20 dialed calls" list (L_z), and total call count (F_n).

Bayes rule [1] of conditional probability is given by Eq. (1).

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \quad (1)$$

where $P(A|B)$ is the *posterior* probability which is the probability of the state of nature being A given that feature value B has been measured. The *likelihood* of A with respect to B is $P(B|A)$ which indicates that other things being equal, the category A for which $P(A|B)$ is large is more "likely" to be the true category. $P(A)$ is called *prior* probability. The *evidence* factor, $P(B)$, can be viewed as a scale factor to guarantee that the posterior probabilities sum to one.

We use this rule to obtain the probability of a number being received given a specific hour of the day, day of the week, current "last 20 dialed calls" list, and total call count, as given by Eq. (2).

$$P(T_n | D_x, H_y, L_z, F_n) = \frac{P(D_x | T_n)P(H_y | T_n)P(L_z | T_n)P(F_n | T_n)P(T_n)}{P(D_x, H_y, L_z, F_n)}, \quad (2)$$

A known issue with the Naïve Bayesian classifier occurs if a particular attribute value doesn't occur in conjunction with every class value in the training data. The attributes in our case are D_x , H_y , and L_z . The class values are the incoming telephone numbers. The computed probability of a number being received at a particular time will be zero if the training data has no instance of that number being received during either the specified hour or the specified day.

A solution to this problem is to start all the call counts in the Hash tables for day-of-week and hour-of-day at one instead of zero and introducing some normalizing factors in the resulting computations.

This is not an issue for the F_n since there must be at least one call count for any seen incoming call. For L_z , this is sort of an issue since only those numbers/contacts that are on the current "last-20-dialed-calls" list are considered. A solution for this case is to assign the lowest call count of the position on the last-20-dialed-calls list (hash table) to those phone numbers that are not on the current last-20-dialed-calls list. Therefore, those numbers that are not on the current last-20-dialed-calls list will have the same probability of being received as the lowest probability of the number on the current list being received. There is also a possibility of one telephone number occupies more than one position on the current last-20-dialed-calls list. In this situation, the highest call count among all positions occupied by that telephone number is assigned to it.

Adopting this approach, we compute the likelihood of a number T_n being received, given $D_x, H_y, L_z,$ and F_n , by Eq. (3).

$$L(T_n | D_x, H_y, L_z, F_n) = \left(\frac{C(T_n D_x) + 1}{C(T_n) + 7} \right) \cdot \left(\frac{C(T_n H_y) + 1}{C(T_n) + 24} \right) \cdot \left(\frac{C(T_n L_z)}{C(L)} \right) \cdot \left(\frac{C(T_n F_n)}{C(T_n)} \right), \quad (3)$$

where $C(T_n D_x)$ is the call count from caller T_n on day D_x ($x = 1, 2, 3, \dots, 7$), $C(T_n H_y)$ is the call count from caller T_n during hour H_y ($y = 0, 1, 2, \dots, 23$), $C(T_n L_z)$ is the call count from caller T_n when T_n 's position on the current last-20-dialed-calls list is L_z ($z = 1, 2, 3, \dots, 20$), $C(T_n F_n)$ is the total call count from caller T_n ($n = 1, 2, 3, \dots, N$, where N is the total number of callers that have made at least one call to the user), $C(L)$ is the total call count of all position on the list (sum of the second column of hash table in Fig. 7), and $C(T_n)$ is the total call count from caller T_n over the whole training data.

4 Performance Analysis

In this section, the CPL is tested against the actual call logs of 20 mobile phone users as described in Section 3. The first two months (approximately 60 days) of call logs are used to train the CPL and the rest of the call logs are assumed to be the future observed call activities to test the performance of the CPL by observing for each call received what position that actual caller has in the predicted list.

Clearly, if the CPL performed perfectly, one would expect the actual caller to be at the top of the predicted list. Generally, such performance is not achievable, but one might expect that the actual caller would tend to appear earlier rather than later in the list.

The overall performance of the CPL based on these 20 users is shown in Fig. 2 where the its accuracy is measured by the average percentage of the actual callers listed within the predicted list as the length of the list varies from 1 to 20 comparing with the accuracy of the conventional “last 20 received calls” list. Figure 8 shows that the CPL outperforms the “last 20 received calls” list with roughly 20% better accuracy.

If there was only one caller, the CPL would always predict the caller correctly. In general, the population of the callers increases (e.g. meeting new friends, signing up for a new phone list, telemarketers gain access to your phone number, etc.). This increasing number of caller population may affect the accuracy of the CPL, i.e. it becomes harder to select a correct number out of a larger sample space.

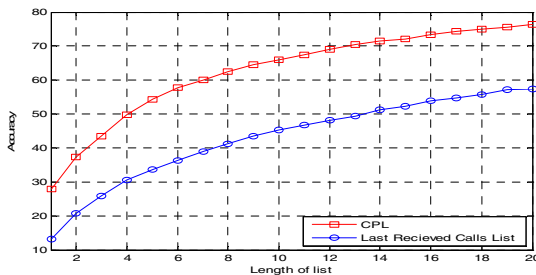


Fig. 2. Overall performance of the CPL comparing to the conventional “Lat 20 Received Calls” list

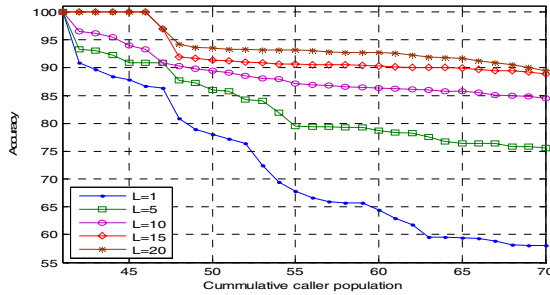


Fig. 3. Relationship between the accuracy of the CPL and the cumulative caller population

Figure 3 shows the relationship between the caller population and the accuracy of the CPL by selecting phone user #20 as an example where the vertical axis represents the accuracy of the CPL, and horizontal axis represents the cumulative caller population which continues to increase from 41 callers to 70 callers. Figure 3 shows that the accuracy decreases dramatically as the caller population becomes larger for different length of the list ($L = 1, 5, 10, 15, 20$). The accuracy drops with relatively higher rate for shorter length of the list as one may expect.

At the same time, the new callers or first-time callers (whose call received for the first time) also degrade the performance of the CPL. This may be an issue for those users who are more social and those who are unfortunately on telemarketers’ lists. This is a voice spam problem which is expected to increase especially in the VoIP networks where the cost of communication is relatively low and with the absurdly large IPv6 address (can support up to 2^{128} addresses).

To demonstrate the impact of the new callers, we examine the accuracy of the CPL without considering the new callers, i.e. if the caller is the first-time caller then it is not taken into account for the accuracy computation. However, after the first call, the caller will be recognized and taken into account for accuracy computation as normal.

It can be seen from Fig. 4 that the accuracy of the CPL is indeed improved about 8% as the new callers are not considered.

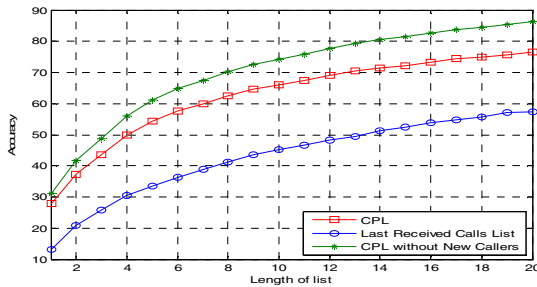


Fig. 4. Overall performance of the CPL without considering first-time callers comparing to the original CPL and the conventional “Lat 20 Received Calls” list

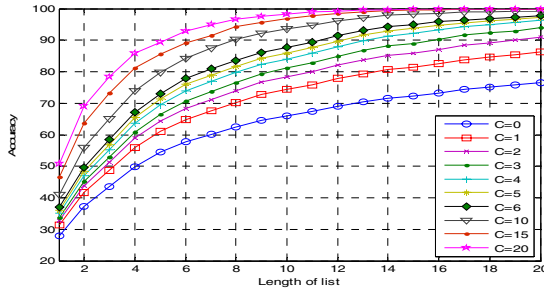


Fig. 5. The impact of the new callers to the accuracy as the criterion of new caller (C) varies from 0 to 20

If we modify our definition or criterion for a new caller by defining a new caller to be a caller who has called C times in the past, then we observe that as variable C increases the accuracy of CPL also increases accordingly, as can be seen in Fig. 5. This tells us that CPL can predict more accurately for the callers whose behaviors have been learned for a longer period of time.

We can further extend the concept of the new callers by using variable C to infer the *social closeness*. It is reasonable to assume that the callers who have made higher number calls to the user are more socially connected to the user. Thus, we can classify callers into two groups based on the number of calls received.

For any given phone user, let \bar{C} be the average number of calls received per caller during one particular time. For any callers who have made less than \bar{C} calls to the user, such callers are classified as *socially distant callers (SDC)* e.g. telemarketers, wrong-number callers, and voice spam, which are normally unwanted calls. On the other hand, for any callers who have made at least \bar{C} calls to the user, such callers are classified as *socially close callers (SCC)* e.g. family members and friends.

$$Caller = \begin{cases} SDC, & C(T_n F_n) < \bar{C} \\ SCC, & C(T_n F_n) \geq \bar{C} \end{cases}, \tag{4}$$

Based on our 20 phone users, the users received an average of six calls per caller during the first two months (learning period). According to Eq. (4), the callers who have made at least six calls are considered socially close callers and the rest of the callers are socially distant callers.

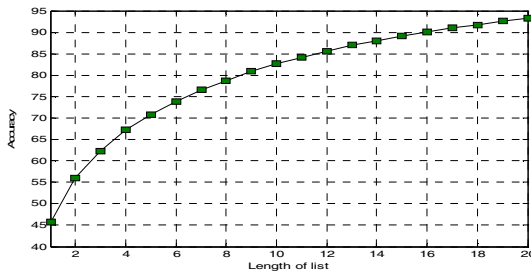
Table 1 shows the accuracy of the CPL for callers who have made at least six calls to the user ($C=6$) at the different length of the list (1, 5, 10, and 20) for each user.

Table 1 shows the comprehensive result which reflects the genuine character of the CPL whose mechanism driven by Bayes rule of conditional probability where the future events conditioned by the past. Hence CPL needs input of historical call logs to learn calling behavior. In fact, it only needs at least six calls for each caller to be an effective predictor. In addition, *SCC* are normally family members and friends who are more desired callers than *SDC* who are normally telemarketers and voice spam.

From Table 1, if the list is only allowed one entry, the CPL would have correctly predicted the socially close callers on average of 40% of the time. If the list has five

Table 1. The performance of the CPL for all 20 users for different length of the predicted list (1, 5, 10, 15, and 20)

Phone User	Accuracy of CPL as length of the list (L) varies (%)				
	$L=1$	$L=5$	$L=10$	$L=15$	$L=20$
1	23.68	60.53	84.21	94.74	100.00
2	16.15	51.55	66.77	83.85	91.93
3	62.00	98.00	100.00	100.00	100.00
4	30.95	92.86	100.00	100.00	100.00
5	42.71	97.92	98.96	100.00	100.00
6	30.42	70.28	91.96	96.50	99.30
7	33.33	100.00	100.00	100.00	100.00
8	39.17	68.66	84.33	93.09	97.24
9	12.90	45.16	77.42	98.39	100.00
10	48.51	90.10	96.04	100.00	100.00
11	10.56	38.73	71.83	90.85	99.30
12	35.71	92.86	100.00	100.00	100.00
13	11.11	35.90	64.96	81.20	87.18
14	74.25	94.31	98.66	99.67	100.00
15	14.29	49.21	76.19	92.06	98.41
16	13.31	45.04	67.99	75.35	82.72
17	68.82	91.25	98.86	100.00	100.00
18	52.28	76.14	89.15	95.44	98.92
19	43.75	69.17	88.75	98.75	100.00
20	73.53	93.38	100.00	100.00	100.00

**Fig. 6.** The performance of CPL as outgoing call predictor (Intelligent Address Book)

entries, the CPL would have correctly predicted the callers 75% of the time. The accuracy would reach 90% for the list of only ten entries.

Since call logs represent human behavior associated with trends and changes over time, thus the accuracy of the CPL can also be impacted by the change of the caller's life schedule because it changes the calling pattern towards the user. For example, your friend changes job from working Monday through Thursday from 8AM to 5PM to working Friday through Sunday from 6PM to 3AM. This major change of your friend's life schedule may result in totally different calling pattern towards you, from receiving several calls at night and on weekends to several calls during the day and on weekdays, for instance. With change of calling pattern of several callers could degrade the performance of the CPL even more.

The concept of CPL can be extended to predicting outgoing calls. For any time the user attempts to make a call (e.g., unlock the keypad, flip up the phone, etc.), a list of the most likely contacts/numbers to be dialed is generated according to computed probability based on call history (day, hour, total call count, and reciprocity). This feature can be envisaged as an “Intelligent Address Book” to reduce the searching time and enable better life synchronization for the phone user. The performance of this Intelligent Address Book is shown in Fig. 6 where it can achieve average accuracy rate of 45%, 70%, and 85%, for the list of one, five, and ten entries, respectively.

5 Conclusion

In this paper, we present a novel concept of the Call Predicted List (CPL) that provides phone user an ability to predict future incoming calls as well as an improvement over the “last received calls” functionality that is often provided on today’s phones and communication clients (e.g. VoIP soft phones). CPL makes use of the user’s call history to build a probabilistic model of calling behavior based on the caller’s calling patterns and reciprocity. The calling behavior model is then used to generate a list of numbers/contacts that are the most likely to be the callers for the next hour. To validate the performance of the CPL, the real-life call logs of 20 mobile phone users are used. The accuracy of the CPL is measured by the percentage of the actual callers listed within the predicted list as the length of the list varies from 1 to 20. The CPL shows 20% improvement in accuracy over the conventional “last 20 received calls” list. In addition, we infer the social closeness from number of calls received as we classify callers into two categories; socially distant callers (e.g. telemarketers, voice spam) and socially close callers (e.g. family members, friends). We believe that socially close callers are more desired callers than socially distance callers. Based on our call logs of 20 phone users, we find that callers who have made at least six calls to the user can be classified as socially close callers for which the CPL accurately predicts 40% if the length of the predicted list is one, 75% if the length is five, and 90% if the length is ten. We also discuss that the accuracy of the CPL can be also impacted by the increase of caller population, new callers, and change of caller’s life schedule. We also show that with a simple modification in input variables, CPL can also be useful for predicting outgoing calls as an “Intelligent Address Book,” by which for any time the user attempts to make a phone call, a list of the likely contacts/numbers to be dialed based on call history is generated. We believe that CPL helps pave the way for future pervasive computing research, which aims to improve quality of life. As our future direction, we will continue to investigate other parameters to characterize and detect the trends/changes in calling behaviors, and explore other prediction techniques to improve the accuracy of the CPL.

Acknowledgements. This work is supported by the National Science Foundation under grants CNS-0627754, CNS-0619871 and CNS-0551694.

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. A Wiley-Interscience Publication, New York (2001)
2. Kolan, P., Dantu, R., Cangussu, J.W.: Nuisance Level of a Voice Call. In: ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) (November 2008) (to appear)
3. Phithakkitnukoon, S., Dantu, R.: UNT Mobile Phone Communication Dataset (2008), http://nsl.unt.edu/santi/data_desc.pdf

OnToContent+QSI 2008 PC Co-chairs' Message

Welcome to the proceedings of the international workshop on Ontology Content and Quantative Semantic Methods (OnToContent+QSI 2008), building on the OnToContent workshop series and augmenting it with new features. This book reflects the issues raised and presented during the workshop.

The workshop was organized and partially funded by

- the Ontology Outreach Advisory OOA (Knowledge Web NoE, FP6-507482). The OOA is devoted to the development of strategies for ontology recommendation and standardization, thereby promoting and providing outreach for verifiable quality ontological content (<http://www.ontology-advisory.org>).
- the MATURE IP (<http://mature-ip.eu>), a large-scale Integrating Project in the field of technology-enhanced learning,
- the SEARCHiN (<http://grid.ucy.ac.cy/SEARCHiN/index.html>) EU Marie Curie actions, developing an advanced method for SEARCHing In a Networked world.
- the Center for Intelligent Computing and Robotics of the Tecnologico de Monterrey, Mexico.

This year, a total of 20 papers were submitted to OnToContent+QSI. Each submission was reviewed by two to four experts. The papers were judged according to their originality, validity, significance to theory and practice, readability and organization, and relevancy to the workshop topics and beyond. This resulted in the selection of 8 papers for presentation at the workshop and publication in these proceedings (40% acceptance rate). We feel that these proceedings will inspire further research and create an intense following. The Program Committee comprised: Ernst Biesalski, Thanasis Bouras, Simone Braun, Christopher Brewster, Michael Brown, Yannis Charalabidis, Ernesto Damiani, Panagiotis Gouvas, Giancarlo Guizzardi, Mohand-Said Hacid, Martin Hepp, Stijn Heymans, Christine Kunzmann, Stefanie Lindstaedt, Tobias Ley, Clementina Marinoni, Alessandro Oltramari, Viktoria Pammer, Paul Piwek, Christophe Roche, Peter Scheir, Miguel-Angel Sicilia, Barry Smith, Armando Stellato, Sergio Tessaris, Robert Tolksdorf, Franky Trichet, Luk Vervenne, Miguel A. Alonso Pardo, Ernesto Damiani, Jerome Euzenat, Sara Garza, Randy Goebel, Adolfo Guzman, Graeme Hirst, Fakhri Karray, Richard Kittredge, Ana Maguitman, Trevor Martin, Antonio Moreno Ortiz, Vivi Nastase, Eduardo Ramirez, Vasile Rus, Elie Sanchez, Juan M. Torres Moreno, Manuel Vilares, and Hugo Zaragoza.

We would like to express our deepest appreciation to the authors of the submitted papers and thank all the workshop attendees and the program committee members for their dedication and assistance in creating our program and turning the workshop into a success. Producing this book would not have been possible without the much appreciated contribution of Pilar Herrero and Anshuman Mukherjee.

Thank you and we hope you enjoy the papers as much as we do.

November 2008

Ramon Brena
Andreas Schmidt, Mustafa Jarrar
Werner Ceusters, Francisco Cantu

Measuring the Benefits of Ontologies

Tobias Bürger and Elena Simperl

Semantic Technology Institute (STI), Innsbruck, Austria
{tobias.buerger,elena.simperl}@sti2.at

Abstract. The technical challenges associated with the development and deployment of ontologies have been subject to a number of research initiatives since the beginning of the nineties. By comparison the economics of ontology engineering remains a poorly exploited field, this underdevelopment having an impact on the adoption of ontology-driven technologies beyond the boundaries of the academic community. The work presented in this paper aims at the alleviation of this situation. We introduce a method for measuring the benefits of ontologies based on a multiple gap model for user information satisfaction analysis. Together with cost models such as ONTOCOM, it can be used to give an account of the economic value of ontologies.

1 Introduction

Ontologies have gained momentum with the emergence of the Semantic Web. Their mainstream adoption – in particular in corporate environments – is, nevertheless, inconceivable in the absence of methods which address the *economic* challenges of ontology engineering in addition to the *technical* and *organizational* ones. Reliable methods to measure and predict the economic value of ontologies are a central component thereof.

The costs of ontology engineering projects are addressed by models such as ONTOCOM [21]. The benefits of using ontologies have also been investigated, however primarily at the level of single applications in terms of special-purpose evaluation criteria [17,16]. What is missing are cross-application methods which can be universally applied to measure or predict these benefits from a technical, but also from a user and an organizational perspective. Such methods would produce results which could be unambiguously interpreted and compared beyond the boundaries of specific technical evaluation frameworks, giving an objective account – together with cost models as that mentioned above – of the added value of ontologies and their applications.

In this paper we propose a method for measuring the benefits of ontologies. The method is generic in the sense that it is not restricted to particular (types of) ontologies or application scenarios thereof. It is based on user information satisfaction analysis, which at its core argues that user satisfaction can be a reliable indicator of the extent to which a particular application meets its objectives, thus allowing to differentiate between more or less successful applications. We

have applied this approach to the ontology engineering field designing a questionnaire which can be used to assess the perceived importance and performance, in other words, the benefits (particular aspects) of ontology-based applications.

2 Benefits of Ontologies

In order to make a statement about the economic value of ontologies, we first need to analyze the benefits they are expected to provide in specific application scenarios or business sectors.

Gruninger and Lee introduced a high-level classification of ontology benefits in [9]. The authors differentiate between three classes of benefits as follows:

- *Communication* between systems, between humans, and between humans and systems.
- *Computational inference*.
- *Reuse and organization of knowledge*.

This classification has been empirically grounded in several studies, among the most recent being the ones by Cardoso [5], and Paslaru and Tempich [4]. According to Cardoso, for instance, ontologies are mostly used to make domain assumptions explicit (70 percent), to enable reuse of domain knowledge (56 percent), or to share a common understanding of the structure of information among people or software agents (37 percent). The usage of ontologies as a means to allow logical inference seems to be less relevant so far.

The previously mentioned classification was further refined at a technical level in [9] and [10] as follows:

- Ontologies can be used for *communication* purposes to
 - Ensure *interoperability* between computer programs (and humans) at data and process level.
 - *Disambiguate* or uniquely identify the meaning of a concepts in a given domain of interest.
 - To facilitate *knowledge transfer* by excluding unwanted interpretations through the usage of formal semantics.
- Ontologies enable *computational inference*, which is in turn useful to
 - Automatically derive implicit facts to enhance traditional *browsing* and *retrieval* technology.
 - Provide an instrument to model domain knowledge independently of the underlying system implementation and enable the *automatic generation of code*.
 - *Spot logical inconsistencies* which potentially indicate modeling errors.
- Ontologies are also means to *structure* and *organize knowledge* in *reusable* artifacts to

- Develop systematic, widely accepted domain descriptions.
- Avoid the costs of new developments whilst potentially increasing the quality of the knowledge models.

A complementary technical analysis of the most prominent application scenarios (and associated expected benefits) of ontologies is provided in [20] (also based on [14,17,26]):

- *Integration*: The ontology provides an integrating environment, an *inter-lingua*, for information repositories or software tools.
- *Semantic search/retrieval*: In this scenario ontologies are used to refine common (keyword-based) search algorithms using domain knowledge in form of subsumption relations or logical constraints. Furthermore they might be used to control the query vocabulary accepted, or to browse the results returned on the basis of domain-specific patterns.
- *Semantic annotation*: In this scenario the goal of the ontology is to provide a controlled vocabulary, as well as a clearly defined classification and browsing structure for the information items within a repository.
- *Software engineering*: The usage of ontologies in the context of software engineering is strongly influenced by the emergence model-driven architectures, which envision to apply them for software verification and validation. A second application is software configuration; here an ontology is used to separate the configuration parameters of a software application from a particular implementation.
- *Knowledge representation*: The ontology is used as a means to formalize the kinds of things that can be talked about in a system or a context.

The roles an ontology might play in these application scenarios can be summarized into four categories:

- *Vocabulary*: The ontology is used as a controlled vocabulary describing the most important concepts of a domain of interest, their properties and their relations to each other. This role is sometimes termed as *inter-lingua* or *lingua-franca*.
- *Formal model*: In contrast to the previous category the ontology is used as a model describing the way concepts in a domain are interconnected and the axioms constraining the meaning and the behavior of the concepts. The formality aspect refers to the usage of a representation language with machine-understandable semantics.
- *Index*: This role characterizes how ontologies are applied as classification structure to index the information items within a repository. The focus is again on the usage of a controlled vocabulary: the ontology pre-defines a set of domain categories, but also on the hierarchical organization of these categories.
- *Filter*: The ontology is applied to refine the results of a specific algorithm, usually in an information retrieval or information management context.

Based on the analysis of ontology benefits presented in this section we have identified several requirements to be fulfilled by the prospective method(s):

1. As most of the ontology benefits cannot be directly quantified, the method should be able to handle intangible benefits.
2. The method should be suitable for application scenarios and business sectors to give full particulars of the benefits of using ontologies.
3. An important impact of the use of ontologies is related to communication. Hence, the method should not rely solely on financial outputs, since this type of outputs are perceived to be less able to capture the full range of communication-related benefits [24].
4. It is generally accepted that an ontology produces added value primarily when it is used in collaboration with other (knowledge) resources as part of a business process. Thus, the method should allow to assess the benefits of an ontology in the context of the application using it.

In the following we give an overview of the most important approaches to IT benefits assessment so as to identify those which are likely to be applicable to the ontology engineering context based on these requirements.

3 Assessing the Benefits of Ontology-Based Applications

A number of methods to assess the benefits of IT are available in related fields such as information systems or business informatics. Most of these methods can be used to deduce comparable figures from tangible, that is directly measurable, and intangible, that is hidden or not obvious, benefits. In terms of the types of outputs produced we can differentiate between financial and non-financial methods [2]. In the first category we count those methods that assess financial value to IT investments by analyzing their cash in- and out-flow. The second category can be further divided into quantitative and qualitative methods, which can be applied complementarily to financial indicators.

We have performed a comprehensive literature study of methods for assessing the value of IT systems [2,6,18,24]. Some of the most important in terms of their industrial impact are:

- *Strategic match analysis and evaluation (SMA)*: SMA [24] is probably the most popular non-financial methods. It tests qualitatively and quantitatively to which extent an IT system supports the overall company strategy by assigning scores to the most important components of the system, which measure the degree to which these components contribute to the achievement of the company's objectives.
- *Value chain assessment (VCA)*: VCA [22] is a scoring and ranking method which measures tangible and intangible value effects using six value factors and four risk factors. The value factors include yield (i.e. value acceleration), strategic value (support existing strategy), competitive advantages, decision support value, minimizing risks, and strategic information system

architecture. The risks include technical, organizational, infrastructure and user-defined risks. The approach is understood as an evolutionary process with the goal to further improve the assessed solution. It is based on user surveys in which users are asked to provide feedback on the value and limitations of an IT solution.

- *Economic assessment (input/output analysis) (EA)*: EA [24] is a theoretical approach to cost/benefit analysis. The method generates a financial output based on a mathematical model that expresses relationships between the inputs and outputs of an IT system. Information about these relationships is captured through expert judgement.
- *User utility assessment (UUA)*: UUA [23] is a method which computes the value of an IT system in quantitative terms (input, processing, and output) based on the frequency with which the system is used.
- *Value added analysis (VAA)*: VAA [11] is a quantitative method which assesses the value of a system to derive its benefits and their relationship to the associated costs. First the proposed functionality and its impact on the business of the company are specified. Then the effects are manually assigned to values, and iterative assessments are made to evaluate if the proposed values have been achieved at certain costs.
- *Critical success factors analysis (CSF)*: CSF is a business term for those properties of a system that are essential to achieve its goals. The associated analysis [25] is a qualitative method which produces a ranked list of critical factors for the success of an IT investment. To do so, one has to identify the goals of a system in relation to the investment, to isolate the tasks, processes, and resources needed to achieve the goals, and to specify tasks which are required to improve the effectiveness of a system.
- *Measuring the benefits of IT innovation (MBITI)*: MBITI [3] is a quantitative method consisting of two parts: a strategic part which is composed of questions about the background and the strategic aspects of an IT investment, and a benefit part which refers to issues of efficiency, effectiveness and performance. The outputs of this method are threefold: estimates of the cost savings of the IT investment, of the relative increase of the effectiveness benefits, and of the significance of non-measurable benefits.
- *Information economics (IE)*: IE [19] is a framework which assesses the enhanced Return on Investment (ROI), the business domain, and the technology domain using a combination of different methods whose joined result is understood as an indicator of the value of an IT investment. ROI is assessed using traditional cost/benefit analysis, value analysis and innovation measurement. The business domain assessment includes methods to assess the in-organizational factors (i.e. strategic match, organizational risk) or competitive factors (i.e. competitive advantage). The technological domain assessment includes strategic investment assessment and risk assessment. The results of the IE analysis are two numerical indicators, the first one showing the total value of an IT investment and the second the risk of failure of implementing the IT investment.

Table 1. Assessment of the surveyed methods

Method	Global Match	Req. 1	Req. 2	Req. 3	Req. 4
Strategic match analysis and evaluation (SMA)	- -	-	yes	yes (quant.)	no
Value chain assessment (VCA)	- -	0	yes	yes (qual.)	yes
Economic assessment (EA)	-	-	yes	no	yes
User utility assessment (UUA)	+ +	+	yes	yes (quant.)	yes
Value added analysis (VAA)	+	+ +	yes	yes (quant.)	yes
Critical success factor analysis (CSF)	+ +	-	yes	yes (qual.)	yes
Measuring the benefits of IT innovation (MBITI)	-	+	no	yes (quant.)	yes
Information economics (IE)	+	+	yes	yes (quant.)	yes
User information satisfaction (UIS)	+ +	+ +	yes	yes (quant.)	yes

- *User information satisfaction analysis (UIS)*: The concept of UIS as coined by Cyert and March [8] is based on the hypothesis that information systems which meet the needs of a user will lead to satisfaction with that system. Many approaches for UIS analysis exist [8,12,13,15,24]. They have been used to measure user satisfaction in terms of (i) attitudes towards an information system, information quality, and effectiveness. User satisfaction is typically captured through questionnaires. Statistical methods are then used to analyze important factors or gaps between the perceived importance and the performance of a system [24].

Table 1 shows how the methods fulfill the requirements introduced in the previous section. Fulfillment of requirements 2 to 4 is indicated by an "yes" or "no", where for the third requirement we additionally mention the type of output, that is, quantitative or qualitative, of the methods assessed as potentially relevant. In case of the first requirement we further take into account how well the corresponding method can handle intangible benefits. This is measured in terms of a five point scale ranging from "-" (no match) to "+ +" (very good match). The value "0" means that the method could cope with intangible benefits provided these could be quantified. The column "Global match" refers to the extent to which the general idea of the methods are compliant with the aim of this research, that is to measure the benefits of ontologies and applications thereof. Again, this indicator is measured through a five point scale.

Most traditional methods that view benefits as a capital investment, strategic impact or that are based on clear performance criteria are not well suited for our case because they do not take intangible benefits into account. These methods include *SMA*, *VCA*, or *EA*. Others like *CSF* account for intangible benefits, however are either based on the global view of a company or are domain related (like e.g. *MBITI*). The most appropriate methods are the ones which are based on the notion of "added value" like *UUA*, *VAA* or *UIS*. *UUA* argues that a

systems which is heavily used is more successful than others which is generally acknowledged not to be an appropriate measure.

Therefore, as a result of this analysis, we selected the method *User Information Satisfaction* as a potential candidate for the measurement of benefits of ontology-based systems. This method is described in more detail in the remaining of this paper.

4 User Information Satisfaction Analysis for Ontologies

User satisfaction can be measured through a comparison of the expectations users have from an information system and the perceived performance of the system expressed in terms of several facets. Several methods can be applied to measure user satisfaction. They can be divided into two categories:

1. *Single-gap models* such as the Miller-Doyle approach [15] compare the perceived importance and the actual performance of an IT system. This information is collected through interviews of a representative user group.
2. *Multiple-gap models* such as the Kim model [13] are particularly useful for assessing how systems are viewed at various stages of their production, implementation and use. Such models measure the gap between the user's expectations and the system designer's interpretations of these expectations, or the quality of the installed system in relation to the experiences of the users.

We propose to approach the measurement of benefits of ontologies using a multiple-gap model. This model is preferred as it allows to capture richer information while also taking into account different target audiences: domain experts, ontology engineers, application end users. These audiences are provided with dedicated questionnaires covering various attributes/facets of an ontology-based system. The UIS analysis is operationalized by applying the following Formula 1, complemented by means for measuring the gaps.

$$UIS = f(Gap_1, \dots, Gap_n, Influential - Factors), n \in \mathbb{N} \quad (1)$$

This formula formally expresses that UIS can be explained by the identified gaps as well as other influential factors, all of which have to be measured through special-purpose metrics. Remenyi et al. [24] suggest to use statistical techniques such as the factor analysis to conceptualize the gaps. Subsequently, correlation and regression analysis could be used to determine how these factors influence the gaps. Influential factors can also influence the size of the gaps; these factors could include user training or administrative support.

A widely-used instrument for collecting information about the user satisfaction are questionnaires in which users are asked to rate the importance and perceived performance of different aspects of a system (cf. Section 4.2). Factor analysis based on the ratings gathered can then reveal the dimensions of user satisfaction. Subsequently, the mean value of the performance ratings of the users

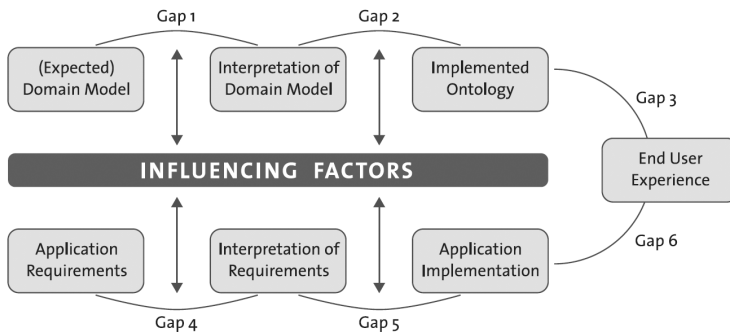


Fig. 1. A Multiple-Gap Model for UIS for Ontology-Based Applications

can be taken as a measure of the perceived performance. Finally, the correlation between the aspects' mean importance and mean performance scores provides an indicator for the overall effectiveness of a system.

4.1 Multiple-Gap Model

The multiple-gap model at the core of our method for measuring ontology benefits is illustrated in Figure 1. It contains two categories of gaps, related to the ontology, and the application using it, respectively:

1. Gap1: This stands for the discrepancy between the expectations of the domain experts, that is, their view of the particular aspects that should be captured through the ontology, and interpretations of these expectations by the ontology engineers.
2. Gap2: This gap accounts for the discrepancy between the ontology engineers interpretation of the domain to be modeled and the actual implementation of the ontology. This gap is typically tested using ontology evaluation methods.
3. Gap3: This gap stays for the discrepancy between the quality of the implemented ontology and the experiences gained by the users when interacting with it.
4. Gap4: This gap, as well as the remaining ones, is application-related. It stands for the discrepancy between the expectations of the user with respect to the functionality of the ontology-based application and the way user requirements are interpreted by the system designers.
5. Gap5: This gap refers to the discrepancy between the interpretation of the user requirements by the designer and the actual implementation.
6. Gap6: The last gap accounts for the discrepancy between the quality of the implemented application and the end user experience.

The gaps were chosen based on the fact that discrepancies between requirements, established quality and user experience influence UIS [13] which accounts for the application related gaps (gaps 4–6). Furthermore it is generally acknowledged

that discrepancies between the expectations of a community on a domain model, the conceptual model and the implemented ontology have implications on the quality of the established ontology which is hypothetically assumed to influence UIS of ontology based systems (gaps 1–2).

4.2 Questionnaire Design

The perceived effectiveness of an IT system can be captured using a questionnaire as briefly sketched in Section 4. In our case this includes questions to measure the perceived importance of different facets (aspects) of a system to assess its effectiveness, questions to measure its performance, and questions to assess the overall user satisfaction with the system. The first part of the questionnaire is based on a taxonomy of facets of ontology-based applications¹. Note that we did not take into account benefits occurring from automation of processes, such as automatic code generation. The facets include attributes for the most frequent uses of ontologies which are based on experiences with ontology-based systems. The participants in surveys which can be based on these attributes will be confronted with questions to rate the importance and actual performance of each facet². The questionnaire will be used to measure the size of the gaps 3–6. Successively ontology metrics will be used to measure the quality of the established ontology (gaps 1–2) in order to demonstrate the influence of the quality of the ontology for UIS. It is expected that after a first factor analysis a stable set of attributes that influence UIS with ontology-based systems will emerge.

5 Outlook and Future Work

In this paper we introduced a model for benefit estimation that can be applied to ontologies based on a multiple-gap model for user information satisfaction analysis. Together with cost estimation methods this model can be used to predict the economic value of ontologies. The model, for sure, leaves room for improvement: After an analysis of the first survey results it could turn out that a combined method of multiple atomic models might be more appropriate. Furthermore the facets of ontology based systems are subject to further discussions. Future work will include carrying out a first survey to collect data about the overall satisfaction and performance of ontology-based systems.

Acknowledgements. The research leading to this paper was partially supported by the European Commission under contract FP6-027122 “SALERO” and FP7-215040 “ACTIVE”.

¹ A mindmap visualizing the facets can be found online:
<http://www.tobiasbuerger.com/mmsurvey/uis/>

² An example of a questionnaire can be found at

<http://www.tobiasbuerger.com/mmsurvey/uis/questionnaire/>

References

1. Aitken, S., Reid, S.: Evaluation of an ontology-based information retrieval tool. In: Proceedings of ECAI 2000 (2000)
2. Andresen, J.: How to select an it evaluation method – in the context of construction. In: Proceedings of the CIB w78 conference, Aarhus School of Architecture (2002)
3. Andresen, J., Baldwin, A., Betts, M., Carter, C., Hamilton, A., Stokes, E., Thorpe, T.: A framework for measuring it innovation benefits. ITcon (2000)
4. Bontas, E.P., Tempich, C.: Ontology engineering: A reality check. In: Proceedings of ODBASE 2006, pp. 836–854 (2006)
5. Cardoso, J.: The semantic web vision: Where are we? IEEE Intelligent Systems, 22–26 (September/October 2007)
6. Carter, C., Thorpe, T., Baldwin, A.: Benefits assessment. ISoCCCrates Deliverable 3, University of Loughborough (1995)
7. Castells, P., Fernández, M., Vallet, D.: An adaptation of the vector-space model for ontology-based information retrieval. IEEE Transactions on Knowledge and Data Engineering 19(2), 261–272 (2007)
8. Cyert, R.M., March, J.G.: Behavioural Theory of the Firm. Prentice-Hall, Englewood Cliffs (1963)
9. Gruninger, M., Lee, J.: Introduction – ontology: different ways of representing the same concept. Commun. ACM 45(2), 39–41 (2002)
10. Hepp, M.: Ontologies: State of the Art, Business Potential, and Grand Challenges. In: Ontology Management: Semantic Web, Semantic Web Services, and Business Applications, pp. 3–22. Springer, Heidelberg (2007)
11. Keen, P.: Value analysis: Justifying decision support systems. MIS Quarterly 5(1), 1–15 (1981)
12. Kim, K.K.: User satisfaction: A synthesis of three different perspectives. The journal of information systems 4(1), 1–12 (1989)
13. Kim, K.K.: User information satisfaction: toward conceptual clarity. In: Proceedings of the International Conference on Information Systems, pp. 183–191 (1990)
14. Leger, A., Nixon, L., Shvaiko, P.: On identifying knowledge processing requirements. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 928–943. Springer, Heidelberg (2005)
15. Miller, J., Doyle, B.: Measuring the effectiveness of computer-based information systems in the financial services sector. MIS Quarterly 11(1) (1987)
16. Oberle, D.: Semantic Management of Middleware (Semantic Web and Beyond: Computing for Human Experience). Springer, Heidelberg (2006)
17. OntoWeb European Project. Ontology-based information exchange for knowledge management and electronic commerce (ontoweb deliverable 2.4) (2003)
18. Parker, M.M., Benson, R.J.: Information Economics - Linking business performance to information technology (1998)
19. Parker, M., Benson, R., Trainor, E.: Information Economics: Linking Business Performance and Information Technology. Prentice-Hall, Englewood Cliffs (1988)
20. Paslaru Bontas, E.: A Contextual Approach to Ontology Reuse: Methodology, Methods and Tools for the Semantic Web. PhD thesis, Free University of Berlin (2007)
21. Paslaru-Bontas-Simperl, E., Tempich, C., Sure, Y.: Ontocom: A cost estimation model for ontology engineering. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273. Springer, Heidelberg (2006)

22. Porter, M.: *Competitive Advantage*. The Free Press (2004)
23. Powell, P.: Evaluation of Information Technology Investments: Business as Usual? In: *Beyond the IT Productivity Paradox*, pp. 151–182. Wiley, Chichester (1999)
24. Remenyi, D., Money, A., Twite, A.: *The effective measurement and management of IT costs and benefits*. Butterworth-Heinemann (1995)
25. Rockart, J.: Chief executives defines their own data needs. *Harvard Business Review*, 1–13 (March-April 1979)
26. SWAD European Project. *Semantic Web applications - analysis and selection (Deliverable SWAD EU-IST-2001-34732)* (2001)

Towards a Human Factors Ontology for Computer-Mediated Systems*

Panagiotis Germanakos^{1,2}, Mario Belk², Nikos Tsianos¹, Zacharias Lekkas¹,
Constantinos Mourlas¹, and George Samaras²

¹ Faculty of Communication and Media Studies,
National & Kapodistrian University of Athens, 5 Stadiou Str, GR 105-62, Athens, Hellas
{pgerman, ntsianos, mourlas}@media.uoa.gr

² Computer Science Department, University of Cyprus, CY-1678 Nicosia, Cyprus
{belk, cssamara}@cs.ucy.ac.cy

Abstract. Adapting to user context, individual features and behavior patterns is a topic of great attention nowadays in the field of Web-based and mobile mediated platforms, such as eTraining, eCommerce, eLearning and so on. A challenge is to design an expressive ontology that is composed of human factors that can be used in any application, whether that is the WWW or any other embedded information system. Based on that ontology, engineers will design and develop personalized and adaptive interfaces and software. This will enable easy access to any content while being sufficiently flexible to handle changes in users' context, perception and available resources, optimizing the content delivery while increasing their comprehension capabilities and satisfaction. Therefore, this paper describes a human factors ontology that has been positively evaluated, called UPPC (User Perceptual Preference Characteristics), and could be used in any computer mediated application for returning an optimized adaptive result to the user.

Keywords: Web Personalization, Adaptation, User Profiling, Cognitive Processes, Ontology, eLearning.

1 Introduction

1.1 Motivation

The usage of computer-mediated systems over the Web has increased rapidly over the past few decades, applied in a variety of domains such as Training, Learning, Commerce and so on. Computer-mediated platforms have been adopted by the mass market much quicker than any other technology / platform over the past century and are currently providing convenience and have changed our lifestyle.

However the plethora of information and services as well as the complicated nature of most Web structures intensify the orientation difficulties, as users often lose sight

* The project is co-funded by the Cyprus Research Foundation under the project EKPAIDEION (#ΠΛΗΠΟ/0506/17).

of their original goal, look for stimulating rather than informative material, or even use the navigational features unwisely. As the e-services sector is rapidly evolving, the need for such Web structures that satisfy the heterogeneous needs of its users is becoming more and more evident [1].

In recent years, there has been a rapid growth in research and experiments that work on personalizing computer-mediated platforms, according to user needs and indeed, the challenges ranging in this area are not few.

Indisputably, the user population is not homogeneous. To be able to deliver quality knowledge, systems should be tailored to the needs of individual users providing them personalized and adapted information.

One of the key technical issues in developing personalization applications is the problem of how to construct accurate and comprehensive profiles of individual users and how these can be used to identify a user and describe the user behaviour. The objective of user profiling is the creation of an information base that contains the preferences, characteristics and activities of the user.

1.2 Proposing a Human Factors Ontology

This paper introduces a new model in the field of Web Personalization, which integrates cognitive, mental and emotional parameters and attempts to apply them on a Web-based learning environment. Our purpose is to improve learning performance in terms of information assimilation and comprehension capabilities and, most importantly, to personalize web content to users' needs and preferences, eradicating known difficulties that occur in traditional approaches.

User Profiling is considered the main filtering element for Web Personalization Systems. Therefore, the main scope of this paper is to further enhance current user profiles by creating an ontology that will be based on a theoretical three-dimensional model incorporating the latter cognitive concepts that has already been proposed by the authors and positively evaluated into the information space [2, 3]. This ontology will contain an optimized series of parameters related to human factors that could be used in any computer-mediated platform in order to return a more enhanced user-centric result.

Using this ontology as the main filtering component we have developed an adaptive Web-based system that could be used to reconstruct (adapt) any content coming from the provider.

Such approach may be proved to be very useful in assisting and facilitating a user to understand better web content and therefore increase his / her satisfaction and navigation performance.

In the remaining sections, we present the theoretical model that the ontology has been based upon, describe its terms and we show the ontology itself. In section 3, we show the implications of each dimension onto the information space. In section 4, we evaluate the ontology's concept in an eLearning domain through a computer-mediated system (Web-based adaptation and personalization system), AdaptiveWeb, since this ontology figures as its core component. Finally, section 5 presents our conclusions and future trends of our work.

2 Describing the UPPC Ontology

We introduce the new component / dimension of the user profile. It contains all the visual attention, cognitive and emotional processing parameters that enhances the user preferences and fulfils the user profile. User Perceptual Preference Characteristics could be described as a continuous mental processing starting with the perception of an object in the user's attentional visual field and going through a number of cognitive, learning and emotional processes giving the actual response to that stimulus, as depicted in Fig. 1, below.

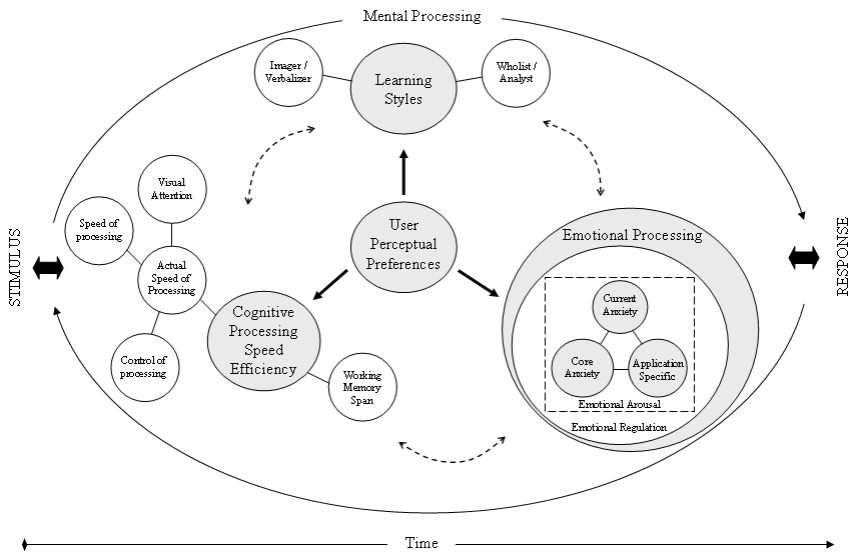


Fig. 1. User Perceptual Preference Characteristics - Three-Dimensional Approach

As it can be observed, its primary parameters formulate a three-dimensional approach to the problem.

These characteristics, which have been primarily discussed in [3], and formulate a three-dimensional approach to the problem of building a user model that determines the visual attention, cognitive and emotional processing taking place throughout the whole process of accepting an object of perception (stimulus) until the comprehensive response to it [4]. The first dimension investigates users' *cognitive style*, the second their *visual and cognitive processing efficiency*, while the third captures their *emotional processing* during the interaction process with the information space [3].

Based on the abovementioned considerations, we present the User Perceptual Preference Characteristics Model that is proposed to identify the main impact of human factors in the information space. Its primary objective is to give a semantic description of each dimension in the UPPC theoretical model as well as the impact in the information space.

2.1 Cognitive Processing Efficiency

The cognitive processing parameters [5, 6] that have been included in our model are:

- i. *control of processing* (refers to the processes that identify and register goal-relevant information and block out dominant or appealing but actually irrelevant information)
- ii. *speed of processing* (refers to the maximum speed at which a given mental act may be efficiently executed), and
- iii. *working memory span* (refers to the processes that enable a person to hold information in an active state while integrating it with other information until the current problem is solved)
- iv. *visual attention* (based on the empirically validated assumption that when a person is performing a cognitive task, while watching a display, the location of his / her gaze corresponds to the symbol currently being processed in working memory and, moreover, that the eye naturally focuses on areas that are most likely to be informative).

We measure each individual's ability to perform control/speed of processing and visual attention tasks in the shortest time possible, with a specific error tolerance, while the working memory span test focuses on the visuospatial sketch pad sub-component [7], since all information in the web is mainly visual.

2.2 Cognitive Style

Cognitive styles represent an individual's typical or habitual mode of problem solving, thinking, perceiving or remembering, and "are considered to be trait-like, relatively stable characteristics of individuals, whereas learning strategies are more state-driven..." [8]. Amongst the numerous proposed cognitive style typologies [9] we favor Riding's Cognitive Style Analysis [10], because we consider that its implications can be mapped on the information space more precisely, since it is consisted of two distinct scales that respond to different aspects of the Web. The imager / verbalizer axis affects the way information is presented, whilst the wholist / analyst dimension is relevant to the structure of the information and the navigational path of the user. Moreover, it is a very inclusive theory that is derived from a number of pre-existing theories that were recapitulated into these two axes.

We prefer the construct of cognitive rather than learning style because it is more stable [11], and to the extent that there is a correlation with hemispherical preference and EEG measurements [12, 8], the relationship between cognitive style and actual mode of information processing is strengthened.

2.3 Emotional Processing

In our study, we are interested in the way that individuals process their emotions and how they interact with other elements of their information-processing system. Emotional processing is a pluralistic construct which is comprised of two mechanisms: emotional arousal, which is the capacity of a human being to sense and experience specific emotional situations, and emotion regulation, which is the way in which an individual is perceiving and controlling his emotions. We focus on these two

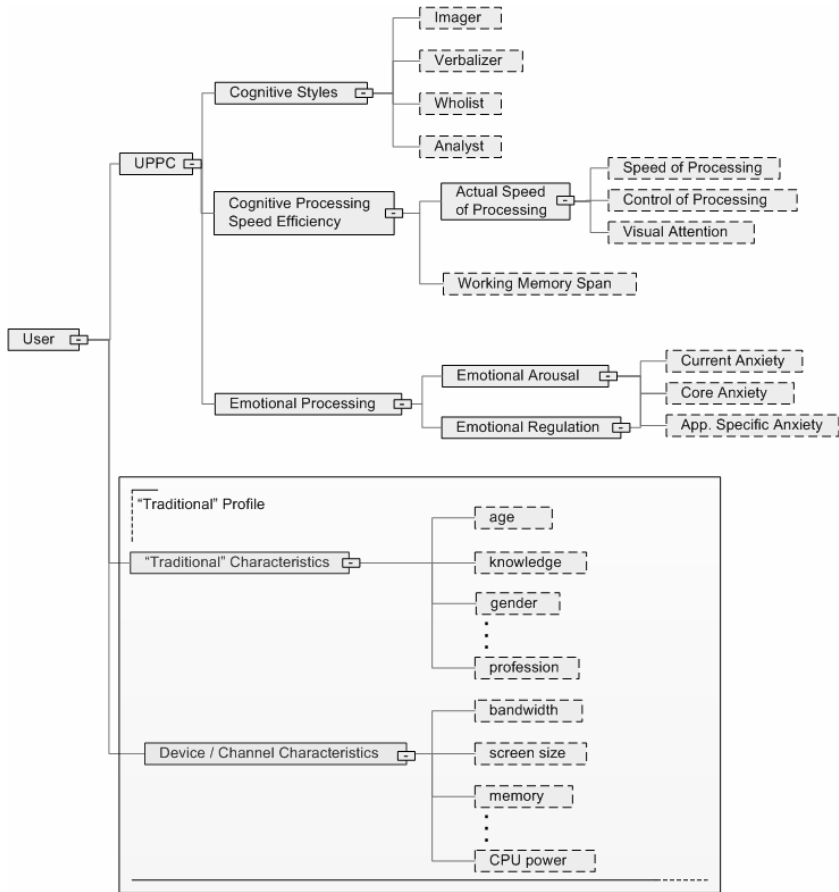


Fig. 2. User Perceptual Preference Characteristics Ontology

sub-processes because they are easily generalized, inclusive and provide some indirect measurement of general emotional mechanisms. These sub-processes manage a number of emotional factors like anxiety boredom effects, anger, feelings of self efficacy, user satisfaction etc. Among these, our current research concerning emotional arousal emphasizes on anxiety, which is probably the most indicative, while other emotional factors are to be examined within the context of a further study.

Anxiety is an unpleasant combination of emotions that includes fear, worry and uneasiness and is often accompanied by physical reactions such as high blood pressure, increased heart rate and other body signals [13] [14].

Accordingly, in order to measure emotion regulation, we are using the cognominal construct of emotion regulation. An effort to construct a model that predicts the role of emotion, in general, is beyond the scope of our research, due to the complexity and the numerous confounding variables that would make such an attempt rather impossible. However, there is a considerable amount of references concerning the role of emotion and its implications on academic performance (or achievement), in terms of efficient learning [15]. Emotional intelligence seems to be an adequate predictor of

the aforementioned concepts, and is a grounded enough construct, already supported by academic literature [16, 17]. Additional concepts that were used are the concepts of self-efficacy, emotional experience and emotional expression [18].

After the presentation of the above findings, we hereafter depict (Fig. 2) the UPPC ontology that uses the main elements of the human factors conceptualization.

The main uses of this ontology [19] are: 1) to enable consistent implementation (and interoperation) of all computer-mediated systems that use human factors as their main filtering element, based on a shared background vocabulary, 2) to play the role of a domain ontology that encompasses the core human factors elements for computer-mediated systems and that can be extended by any other individual or group.

As we will see later, we evaluated the UPPC ontology’s concept in an eLearning domain. Although the results are really encouraging for the validity and integrity of the relation within and between the UPPC model dimensions, this model can only be considered as a proposal. Main goal is to initiate and drive this research to a concrete human factors ontology that can be used in any computer-mediated system.

3 Relating the UPPC Ontology with the Information Space - A High Level Correlation Diagram

For a better understanding of the three dimensions’ implications and the UPPC ontology as well as their relation with the information space a diagram that presents a high level correlation of these implications with selected tags of the information space (a code used in Web languages to define a format change or hypertext link) is depicted

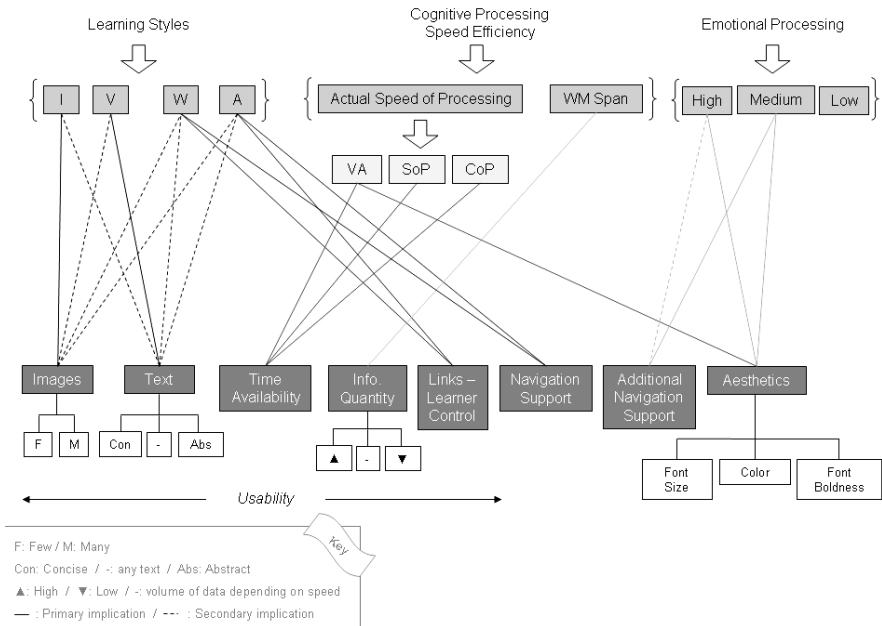


Fig. 3. Data - Implications Correlation Diagram

in Fig. 3. These tags (images, text, information quantity, links - learner control, navigation support, additional navigation support, and aesthetics) have gone through an extensive optimization representing group of data affected after the mapping with the implications. The main reason we have selected the latter tags is due to the fact that represent the primary subsidiaries of a Web-based content. With the necessary processing, mapping and / or alteration we could provide the same content with different ways (according to a specific user's profile) but without degrading the message conveyed.

The particular mapping is based on specific rules created, liable for the combination of these tags and the variation of their value in order to better filter the raw content and deliver the most personalized Web-based result to the user. As it can be observed from the diagram below each dimension has primary (solid line) and secondary (dashed line) implications on the information space altering dynamically the weight of the tags.

Henceforth, with regards to the learning style, the number of images (few or many) for example to be displayed has a primary implication on imagers, while text (more concise or abstract) has a secondary implication. An analyst may affect primarily the links - learner control and navigation support tag, which in turn is secondary affected by high and medium emotional processing, while might secondary affect the number of images or kind of text to be displayed, consequently. Actual speed of processing parameters (visual attention, speed of processing, and control of processing) as well as working memory span are primarily affecting information quantity. Eventually, emotional processing is primarily affecting additional navigation support and aesthetics, as visual attention does, while secondary affects information quantity.

Additionally, since emotional processing is the most dynamic parameter compared to the others, any changes occurring at any given time are directly affecting the yielded value of the adaptation and personalization rules and henceforth the format of the content delivered.

The intended impact of this research is the future initiation of the use of human factors in the information space. In this respect, the ontology is intended to become the basis for a future core human factors ontology in the domain of computer-mediated systems.

4 Evaluation

The UPPC ontology has been evaluated in the eLearning domain through a computer-mediated application (Web-based adaptation and personalization system), AdaptiveWeb, since it figures as its core component. A total number of five hundred BETA testers participated in the experimentation phases. The concept of the ontology has been proven effective and efficient not only regarding the relation within and between its various human factor elements but also in respect to the actual output data gathered which reveals that the whole approach turned out to be initially successful with a significant impact in the Personalization and Adaptation Procedure.

More specifically we have implemented a number of experiments in controlled environments with regards to the three dimensions giving match and mismatch environments, depending on the factor we were controlling each time. Our main hypothesis was that students in matched environment perform better than those in mismatched conditions. The initial evaluative results were really encouraging for the

future of our work since we found that in many cases there is high positive correlation of matched conditions with performance, as well as between the dimensions of the various factors of our model. This fact demonstrates the effectiveness of incorporating human factors in Web-based personalized environments [20].

Initial indications show a significant correlation between the Cognitive Processing factors used, not only amongst them but also with regards to the performance of the subjects. Subjectively, with regards to the cognitive learning styles it has been shown that Imagery, Verbalizers, Wholists and Analysts are greatly performed in environments of their type whereby it has been identified significant correlation with the second dimension and more specifically with the Working Memory Span parameter (Working Memory Capacity is more important for Analysts than Wholists).

Regarding the second dimension, Cognitive Processing Speed Efficiency results have preliminary showed that could be used effectively in controlled mostly learning environments where time span / availability is an issue. With regards to the Working Memory, high significance has been found to low Working Memory subjects whereby broken content is considered necessary for their better comprehension of the Web-based content.

Regarding the third dimension, it has been proven that the medium anxiety level of the subject is more beneficial for their learning performance. For users with high anxiety levels it has been shown that the aesthetics and extra navigation support are considered necessary assistive tools and techniques for their adequate comprehension of the content. Furthermore the emotional regulation of a subject acts reversely with the trait anxiety (mostly with the application specific than the core) and current anxiety. It has been preliminary also found a noteworthy correlation between the anxiety levels and the CPSE of the user. More specifically, users with high CPSE have shown decreased levels of application specific anxiety.

It has to be mentioned that this is a high level preliminary analysis that however indicates initial validity of the proposed model and the positive impact of the AdaptiveWeb System.

In due time further experiments will be conducted enabling us to analyze in more depth the parameter significance and correlation with regards to the information space and therefore prove validity of our research argument and the proposed UPPC ontology.

5 Conclusions and Future Trends

The basic objective of this paper was to present a conceptualization of a human factors ontology, namely UPPC, for computer-mediated systems.

It has been attempted to approach the theoretical considerations and technological parameters that can provide the most comprehensive user profiling, under a common filtering element (User Perceptual Preference Characteristics), supporting the provision of the most apt and optimized user-centered Web-based result. We further underpinned the significance of the user profiling, introducing the comprehensive user profiling, which incorporates intrinsic user characteristics such as user perceptual preferences.

Furthermore, a data - implications correlation diagram has been identified showing the impact of human factors and more specifically of the UPPC parameters onto the information space.

An evaluation with approximately 500 users has been conducted with the results to be highly promising and encouraging for the continuation of our research, since using the proposed human factors ontology as a main filtering element of a computer-mediated system can really increase students' academic performance.

Future and emerging trends include further analysis and testing of the current ontology in different domains and contexts. A more detailed analysis of the current model as well as the relationship between its different sub-dimensions; further investigation of constraints and challenges arise from the implementation of such issues on mobile devices and channels; study on the structure of the metadata coming from the providers' side, aiming to construct a Web-based personalization architecture that will be based on the UPPC ontology and will serve as an automatic personalization filter.

References

1. Germanakos, P., Samaras, G., Christodoulou, E.: Multi-channel Delivery of Services - the Road from eGovernment to mGovernment: Further Technological Challenges and Implications (2005)
2. Germanakos, P., Tsianos, N., Lekkas, Z., Mourlas, C., Samaras, G.: Realizing Comprehensive User Profile as the Core Element of Adaptive and Personalized Communication Environments and Systems. *The Computer Journal*, Special Issue on Profiling Expertise and Behaviour (2008)
3. Germanakos, P., Tsianos, N., Lekkas, Z., Mourlas, C., Samaras, G.: Capturing Essential Intrinsic User Behaviour Values for the Design of Comprehensive Web-based Personalized Environments. *Computers in Human Behavior Journal*, Special Issue on Integration of Human Factors in Networked Computing (2007)
4. Germanakos, P., Tsianos, N., Mourlas, C., Samaras, G.: New Fundamental Profiling Characteristics for Designing Adaptive Web-based Educational Systems. In: *Proceeding of the IADIS International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2005)*, Porto, December 14-16, pp. 10-17 (2005)
5. Demetriou, A., Efkliides, A., Platsidou, M.: The architecture and dynamics of developing mind: Experiential structuralism as a frame for unifying cognitive development theories. *Monographs of the Society for Research in Child Development* 58 (Serial No. 234), 5-6 (1993)
6. Demetriou, A., Kazi, S.: Unity and modularity in the mind and the self: Studies on the relationships between self-awareness, personality, and intellectual development from childhood to adolescence. Routledge, London (2001)
7. Baddeley, A.: Working Memory. *Science* 255, 556-559 (1992)
8. McKay, M.T., Fischler, I., Dunn, B.R.: Cognitive style and recall of text: An EEG analysis. *Learning and Individual Differences* 14, 1-21 (2003)
9. Cassidy, S.: Learning Styles: An overview of theories, models, and measures. *Educational Psychology* 24(4), 419-444 (2004)
10. Riding, R.: Cognitive Style Analysis - Research Administration. *Learning and Training Technology* (2001)
11. Sadler-Smith, E., Riding, R.J.: Cognitive style and instructional preferences. *Instructional Science* 27(5), 355-371 (1999)
12. Glass, A., Riding, R.J.: EEG differences and cognitive style. *Biological Psychology* 51, 23-41 (1999)

13. Kim, J., Gorman, J.: The psychobiology of anxiety. *Clinical Neuroscience Research* 4, 335–347 (2005)
14. Barlow, D.H.: *Anxiety and its disorders: The nature and treatment of anxiety and panic*, 2nd edn. The Guilford Press, New York (2002)
15. Kort, B., Reilly, R.: Analytical Models of Emotions, Learning and Relationships: Towards an Affect-Sensitive Cognitive Machine. In: *Conference on Virtual Worlds and Simulation, VWSim 2002* (2002), <http://affect.media.mit.edu/projectpages/lc/vworlds.pdf>
16. Goleman, D.: *Emotional Intelligence: why it can matter more than IQ*. Bantam Books, New York (1995)
17. Salovey, P., Mayer, J.D.: Emotional intelligence. *Imagination, Cognition and Personality* 9, 185–211 (1990)
18. Schunk, D.H.: Self-efficacy and cognitive skill learning. In: Ames, C., Ames, R. (eds.) *Research on motivation in education. Goals and cognitions*, vol. 3, pp. 13–44. Academic Press, San Diego (1989)
19. Jarrar, M.: Towards Effectiveness and Transparency in e-Business Transactions, An Ontology for Customer Complaint Management. In: *Semantic Web Methodologies for E-Business Applications*. Idea Group Inc. (2007)
20. Tsianos, N., Germanakos, P., Lekkas, Z., Mourlas, C., Samaras, G.: Evaluating the Significance of Cognitive and Emotional Parameters in e-Learning Adaptive Environments. In: *Proceedings of the IADIS International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2007)*, Algarve, Portugal, December 7-9, 2007, pp. 93–98 (2007)

Ontology Development in Collaborative Networks as a Process of Social Construction of Meaning

Carla Pereira^{1,2} and António Lucas Soares^{1,3}

¹ Instituto de Engenharia de Sistemas e Computadores do Porto (INESC Porto),
Campus da FEUP, Rua Dr. Roberto Frias, 378, 4200 - 465 Porto, Portugal

² Escola Superior de Tecnologia e Gestão de Felgueiras – Instituto Politécnico do Porto,
Rua do Curral, Casa do Curral, Margaride, 4610-156, Felgueiras, Portugal

³ Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, 4200-465,
Porto, Portugal

csp@inescporto.pt, als@fe.up.pt

Abstract. This paper proposes a new method to support the collaborative construction of semantic artifacts in an inter-organizational context. It aims at being applied, in particular, in the early phases of ontology development. We share the view that the development of semantic artifacts in collaborative networks of organizations should be based on a continuous construction of meaning, rather than pursuing the delivery of highly formalized accounts of domains. For that, our research is directed to the application of cognitive semantics results, specifically by developing and extending the Conceptual Blending Theory to cope with the socio-cognitive aspects of inter-organizational ontology development. Besides the outline of the method, we analyze the main problems and gaps in current ontology development methods regarding collaboration and negotiation in early development phases.

Keywords: conceptualization, social construction of meaning, ontology development, collaborative networks, conceptual blending.

1 Introduction

This research work addresses the problems raised by information and knowledge sharing in the context of short life-cycle collaborative networks¹. Although there is an increasing number of semantic tools and resources available for the companies to use in everyday business activities, problems in establishing a common conceptualization of a given reality arise in two flavours: (i) notwithstanding the evolution of semantic technologies, it is virtually impossible to establish a priori comprehensive and complete semantic artifacts that account for all the possible variations in business situations

¹ Collaborative networks are understood here as networks of organizations -typically SMEs- that establish loose, long-term ties (social, technological, other) that enable a swift formation of inter-organizational teams to pursue a business opportunity (some call this a virtual enterprise). The market and economic trends show that, in general, these business opportunities will be more frequent and shorter in the future [19].

and contexts (which are more and more dynamic); (ii) in spite of all the standardization efforts, there is a kind of “social resistance” in accepting semantically oriented standards (viewed as “grand narratives” of a domain). A good example of this is the construction industry, where an enormous effort and money has been spent in standard terminologies, vocabularies, thesaurus, ontologies, etc. with results well behind the expected.

As clearly argued in [1] about the role of a “socio-semantic web”, we need to go beyond of approaches that provide an high level of “automation of the meaning” with formal ontologies built by ontologists and processed by software agents using automated inferences. Instead, we need to address situations where human beings are highly required to stay in the process, interacting during the whole life-cycle of applications, for cognitive and cooperative reasons [1]. A class of these situations is the setup and deployment of collaborative tasks involving information and knowledge sharing in the context of collaborative networks.

In the scientific context, research on ontology engineering addressed poorly the above problems. From a comprehensive review of the state-of-the-art we concluded that, in general, not only collaboration is superficially considered, as the view of meaning as socially constructed is almost discarded. In particular, research on the early phase of ontology construction (a set of tasks resulting in an informal specification of a conceptualization), has been scarce. Current knowledge about the early phases of ontology construction is insufficient to support methods and techniques for a collaborative construction of a conceptualization. However, the conceptualization phase is of utmost importance for the success of the ontology. But it is in this phase that a social presence is needed as it requires an actor to predict reliably how other members of the community will interpret the conceptual representation just based on its limited description. By incorporating the notion of semantics into the information architecture, we, thus transform the users of the system themselves into a critical part of the design.

Our view is that ontology engineering needs a “socio-cognitive turn” in order to generate tools that are really effective in coping with the complex, unstructured, and highly situational contexts that characterize a great deal of information and knowledge sharing in businesses collaboration. Our stance is thus a socio-semantic [1] one as we believe that the development of semantic artifacts in collaborative networks of organizations should be based on a continuous construction of meaning, rather than pursuing the delivery of highly formalized accounts of domains. This is so because of the volatility of business opportunities and the setup and dismissing of partnerships in collaborative networks.

Our research is thus directed towards the application of cognitive semantics results in the creation of artifacts acting as socio-technical devices supporting the view that meaning socially constructed through collaboration and negotiation. The first line of this research work deals with the application and extension of the Conceptual Blending Theory (CBT) [3] to the realm of collaborative semantic tools. The research questions of this line of work are: “how to extend the conceptual blending theory to accommodate contexts of organizational action and interaction?” and “how to operationalize the theory of extended conceptual blending in a practical method supported by a computer-based tool?” In this paper we describe the first version of a method based on CBT aimed at supporting the co-construction of the informal specification of a conceptualization by an inter-organizational team formed out of a collaborative network.

The practical application of our approach is to support the co-construction of semantic artifacts by groups of social actors placed in organizational contexts interacting towards a set of common objectives. Simple examples these artifacts are the creation of a common taxonomy (or ontology) for classifying and retrieving content from an inter-organizational portal, the creation of specific terminological accounts to serve as conceptual references in project tasks, or the specification of ontologies for systems interoperability.

2 A Method to Support a Collaborative Conceptualization Process

2.1 Cognitive Semantics and the Conceptual Blending Theory

The relation between cognitive semantics and knowledge representation is better understood by considering the four principles that collectively characterize a cognitive semantics approach: 1/ the conceptual structure is embodied: The nature of conceptual organization arises from bodily experience, in other words, cognitive semanticists set out to explore the nature of human interaction with an awareness of the external world, and to build a theory of conceptual structure that is consonant with the ways in which we experience the world; 2/ the semantic structure is the conceptual structure: Semantic structure (the meanings conventionally associated with words and other linguistic units) is equated with concepts; 3/ meaning representation is encyclopedic: this means that words do not represent neatly packaged bundles of meaning (the dictionary view), but serve as ‘points of access’ to vast repositories of knowledge relating to a particular concept or conceptual domain; 4/ meaning construction is conceptualization: the language itself does not encode meaning. Instead, as we have seen, words (and other linguistic units) are only ‘prompts’ for the construction of meaning. According to this view, meaning is constructed at the conceptual level: meaning construction is equated with conceptualization, a dynamic process where linguistic units serve as prompts for an array of conceptual operations and the recruitment of background knowledge.

Our proposal to support a collaborative process of conceptualization (e.g., a domain) is founded on cognitive semantics, specifically on the Conceptual Blending Theory [3]. CBT accounts for the emergence of meanings by adopting the view that meaning construction involves emergent structure, i.e., conceptual integration is more than the sum of its component parts. An integration network is thus a mechanism for modeling how emergent meaning might come about, accounting for the dynamic aspects of meaning construction.

CBT representation gives rise to complex networks by linking two (or more) input spaces by means of a generic space (see figure 1). The generic space provides information that is abstract enough to be common to all the input spaces. Elements in the generic space are mapped onto counterparts in each of the input spaces, which motivate the identification of cross-space counterparts in the input spaces. A further space in this model of integration network is the blended space or blend.

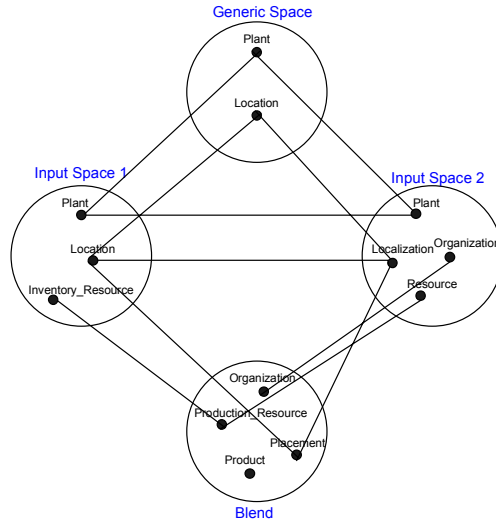


Fig. 1. The network model of Conceptual Blending

This is the space that contains new or emergent structure: information that is not contained in either of the inputs. The blend takes elements from both inputs, but goes further on providing additional structure that distinguishes the blend from either of its inputs. In other words, the blend derives a structure that is contained in neither input.

In CBT, there are three component processes that produce an emergent structure [3]: (1) composition; (2) completion; and (3) elaboration. The first one involves the composition of elements from separate input spaces. In the example of figure 1, composition brings together the concept INVENTORY_RESOURCE of input space 1 with the concept RESOURCE of input space 2 resulting in PRODUCTION_RESOURCE in the blend. Similarly, PLACEMENT in the blend composes the elements projected from the input space 1, LOCATION, with those projected from the input space 2, LOCALIZATION. The second process, completion, involves schema induction [3]. Schema induction involves the unconscious and effortless recruitment of background frames and these complete the composition. Without the structure provided by the strategic frame, we would lose the central inference emerging from the blend. This process of schema induction is called ‘completion’ because structure is recruited to ‘fill out’ or complete the information projected from the inputs in order to derive the blend. Finally, elaboration is the processing that produces the structure unique to the blend. This process is also called running the blend. For example, the concept PRODUCT in the blend results from this process. CBT has more to say about these constitutive processes but further description is outside the scope of this paper (please see [3]).

2.2 Using CBT to Support a Collaborative Conceptualization

CBT seems to have a great potential as foundation for a comprehensive method and associated tools supporting collaborative conceptualization processes in inter-organizational

settings. Although our long-run goal is to extend CBT towards a theory of organizationally extended conceptual blending, a first iteration of the approach can already be described. The following is assumed: (1) a collaborative network has been formed and its goals and mission are defined and understood by all members; (2) a common ontology with certain goals and to be used in a given time-frame has to be developed; (3) each organization has a representative in a “network team” in charge of developing the ontology. The common conceptualization (remember that it stands for “informal specification of the conceptualization”) regarding given domains, processes and tasks, is the first important collective task to undertake by this team; (4) a common conceptualization (as the first step to the ontology construction) is to be collaboratively created through explanation, discussion and negotiation. This approach is only feasible with the support of a sophisticated tool that facilitates and manages all the process.

The method we propose establishes the following steps: (1) each organization has assigned one or more input spaces (only one input space is considered here, for the sake of simplicity); (2) each organization represents its conceptualization proposal through the input space; simultaneously, the organization share the information and knowledge sources (e.g., URLs, documents and files) which allow for the correct understanding of its conceptualization proposal; no specific knowledge representation technique is proposed, but it is important that it has a graphical nature (e.g., concept maps or topic maps); (3) by some manual or automated (or something in between) process, a generic conceptualization is generated (generic space); the common conceptual structure in the generic space should be generic enough to be tacitly accepted by all the team members with minimum negotiation; (4) considering the “counterpart” elements (concepts of the input spaces subsumed by concepts of the generic space), the process of creating the blend space is started using selective projection; based on the input spaces, strategic frame, generic space and documentation available in the input spaces (called background information), the blend is “run” to obtain new proposals; (5) new conceptual structures proposed in the blend space are object of negotiation; the concepts for which consensus exists are represented in (“copied” to) the generic space; situations that justify “backward projection” to the input spaces and their modification are analyzed (this analysis will be performed by users, after obtaining consensus) then, the emergent blend structure is validated (confirm or eliminate new concepts that raise in the blend); (6) if input spaces modification takes place, the method should resume at step 4; however, is not necessary the creation of a new blend space; (7) when all participants manifest their agreement with the conceptualization represented in the generic space, the method instance is finished.

Summarizing, at the end of the process the generic space contains the collective conceptualization, the blend was used during the negotiation process with the goal to improve, enrich and mainly helping in obtaining consensus (proposing new concepts, modifying, improving or eliminating concepts, in the input spaces, that during the negotiation process showed to be incompatible with the business context).

This method may also be used by each organization to support the creation of its input space, which allows us to say that it can result in the presence of multiple blendings. It is important to reinforce that in a collaborative and social process, the validation/agreement achievement requires that each organization indexes to its input space the sources of information which lead to the input spaces creation and justify the proposal content and structure.

2.3 An Illustration

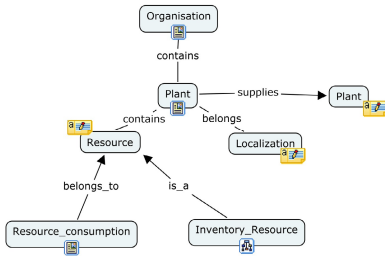
Consider a project team formed out of a R&D international project consortium in the automotive area. One of the tasks of this team is, early in the project plan (3 years project duration), to define an ontology to support several aspects of the project: terminological reference, conceptual clarification, information retrieval, etc. The ontology relates mostly to the project domains, but also to organizational processes and tasks (DPT ontology). Let's consider the development of a part of the ontology related to the sub-domain "Production Network Structure". Each partner organization conceptualizes this domain. Due to individual and organizational differences, mainly in the educational/professional backgrounds and organizational practices and culture, differences in the understanding of apparently common conceptual structures will appear. These differences are even more noticeable when the collaborative working context is one of innovation: new conceptual structures or overloading of existing ones can lead to conceptual confusion. Thus, the "Production Network Structure" sub-domain contains concepts which do not change frequently and that describe the master data of the network structure. The several steps and their results are described below (see figure 2).

Step 1: Strategic Frame Definition

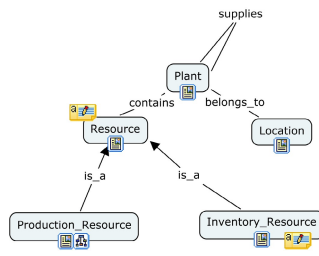
Network Structure Strategic Frame Definition: A conceptual structure that describes the master data of the automotive production network included in the interoperability model. **Goal:** definition of the automotive production network concepts. **Step 2:** Each organization creates an input space where its proposal is presented for the domain conceptualization. To guarantee a correct understanding of each organizational proposal, it is necessary linking to each proposal the "information and knowledge resources" that document its rationale (see figure 2-a and 2-b). **Step 3:** This is a semi-automatic process, based on the concepts and their associated "information and knowledge resources" are identified some similarities and "counterpart" elements. Others are identified by the participants during the negotiation process. It can be easily understood that was considered as "counterpart" elements the concepts "Plant", "Resource", "Inventory_Resource" and "Location". For "Location" and "Localization" in the input spaces, the participants reached agreement that the best term to use is "Location" (see figure 2-c). **Step 4:** "Run the blend" to obtain new proposals. Blend is built based on the input spaces, the strategic frame, generic space and the documentation available as background information for generic and input spaces creation (this is an automatic process).

For example, the blend proposes a new concept ("Placement") as result of the selective projection of "Location" and "Localization". Suggests a new concept ("Product") and a new organization for the "Production_Resource" and "Resource_consumption" concepts (see figure 2-d). Step 5, 6 and 7: After the negotiation process is finished, a shared conceptualization is reached (presented in the generic space). The negotiation process is based on the analysis of the blend proposal, the inputs spaces and in the generic space. As result of the analysis, a new concept ("Feature") is proposed as well as a new connection between "Product", "Resource_consumption" and "Production_Resource" (see figure 2-e).

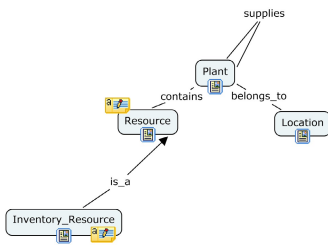
a) Step 2: Input Space 1



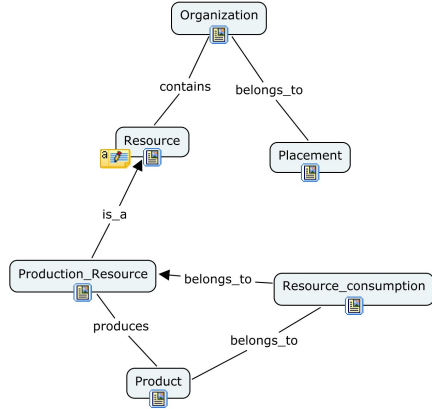
b) Step 2: Input Space 2



c) Step 3: Generic Space



d) Step 4: "Run the blend" (Blend Space)



e) Step 5, 6, and 7: Shared conceptualization (Generic Space)

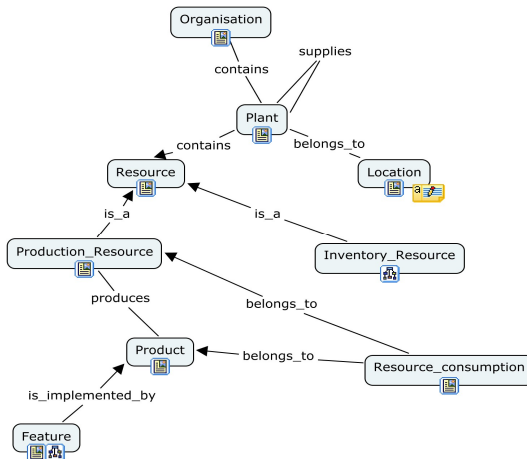


Fig. 2. Illustration of the CBT use for collaborative ontology conceptualization

3 Related Work

The work on ontology engineering methodologies has been extensively discussed and compared in [5], [6], [7] and [8]. We can conclude that the ontology engineering field has laid a lot of emphasis on the “specification of the conceptualization” as an engineering task. Nevertheless, the early phases of the ontology development life-cycle have been poorly addressed. In particular, the importance of the social processes involved in the formation of a collective conceptualization (e.g., of a domain) has not been recognized. The supports to the collaborative processes in ontology construction were thus the focus of our review of related work. We present here an abridged analysis of the research work dealing more directly with collaboration in ontology development (see also [10], [12], [13], [14], [15] and [16] for a complete account of those approaches).

In the “Knowledge Mediation Procedure” [17] the authors present several interesting points: the need to establish a shared conceptualization, the need to consider all of the users’ perspectives, the advantages to use automatic thesaurus generation tools during the knowledge acquisition and ontology mining techniques that can be helpful to reach an agreement and the need of documentation construction in the final of the process. However, experimental evaluations should therefore be considered as a crucial, the authors only present one simple empirical study. In any way, the fact of the process of ontology engineering is also conceived as social process it can be of great value for the conceptualization phase. This is the only approach that focuses explicitly the importance of social presence in the ontology specification phase. It is a very interesting approach, but some questions arise. There is not an explicit step in the approach that accounts for the individual conceptualizations brought by each participant and how these are represented and visualized. There is not any clear theory or even guidelines supporting the negotiation process. An empirical study was carried out aiming at validating all the procedure phases in a scenario closer to the real application. It focus only on the integration phase and lacks information regarding the use of the tools proposed by the method. The study seems also to be on the simplistic side.

In AKEM [18], although the authors refer that there is a large involvement of the users, it is only noticeable in the validation activity. It is also claimed that AKEM puts emphasis on the consensus construction and communication in the ontology development process (based on the DOGMA approach [10]). It is not clear how this process is driven and explored. The DOGMA analysis reveals that the meaning negotiation process is their key research focus at the moment. The authors also claim that AKEM is worked out for the team to organize itself in an iterative agile development life cycle, with emphasis on requirements determination and semantic scoping. We were not able to identify the specific methods and techniques that make this approach “agile”. The result of the knowledge specification is very formal, in our opinion difficult for the users’ participation and validation.

HCOME [11] claims to be a human-centered ontology engineering methodology. However, there are several points that are not clear enough in order to assess this claim. Negotiation is centered in the so called “argumentation dialogues” but the way these support consensus building is not clear-cut. For the conceptualization phase, the authors do not propose a specific approach, suggesting that any existing approach or

combination of approaches could be used. This way, the approach bypasses one of the crucial aspects of collaborative ontology development.

In conclusion, although we have exhaustively reviewed the scientific literature on ontology development, we only found out that a few research works recognizing the importance of supporting the collective construction of a conceptualization. This is in our opinion the cornerstone of any ontology engineering method. Other particular questions come out from this review: (i) the importance of representational tools and user interfaces for interacting with knowledge representations are generally underestimated; (ii) negotiation and consensus building regarding the conceptualization content has not been a priority either; there are a few proposals that claim to support the process of reaching consensus or agreements, but only one addresses the issue of what conceptual content should be included in the shared conceptualization; (iii) the reutilization of existing ontologies is an obvious requirement; nevertheless, there is not any approach that integrates reutilization with the conceptualization building in a systematic way.

4 Conclusions and Future Work

As stated in the beginning of the paper, current knowledge about the early phases of ontology construction is insufficient to support methods and techniques for a collaborative construction of a conceptualization. However, the conceptualization phase is of utmost importance for the success of an ontology, in particular an inter-organizational one. In this paper we outlined a method founded on the conceptual blending theory, that we believe will be an important contribution for the appearance of a new generation of tools to support the co-construction of semantic resources. Short term further work include the following: (1) to select a visual representation technique to support the construction of the input spaces, generic space and blend space, as for example, conceptual maps or topic maps; (2) to define the rules, methods and algorithms that will support the construction of the blend space; (3) to refine the method to create the initial conceptualization in the generic space from the initial input spaces. To complete the first full version of the method, a procedure and algorithm to “run the blend” (step 4 in the method) must be devised. Nevertheless this should probably be developed after the development of the organizational extensions to CBT. This is our current work.

References

1. Cahier, J.-P., Zaher, L.H., Leboeuf, J.-P., Guittard, C.: Experimentation of a socially constructed “Topic Map” by the OSS community. In: Proceedings of the IJCAI 2005 workshop on Knowledge Management and Ontology Management (KMOM), Edimbourg (2005)
2. Evans, V., Green, M.: Cognitive Linguistics: an introduction. Edinburgh University Press (2006)
3. Fauconnier, G., Turner, M.: Conceptual Integration Networks. *Cognitive Science* 22(2), 133–187 (1998)

4. Gruber, T.R.: A translation approach to portable ontology specifications. *Knowledge Acquisition* 5(2), 199–220 (1993)
5. Corcho, O., Fernández-López, M., Gómez-Pérez, A.: Methodologies, tools and languages for building ontologies: where is their meeting point? *Data Knowl. Eng.* 46(1), 41–64 (2003)
6. Fernández-López, M., Gómez-Pérez, A.: Overview and analysis of methodologies for building ontologies. *The Knowledge Engineering Review* 12(2), 129–156 (2002)
7. López, M., Gómez-Pérez, A., Sierra, J., Sierra, A.: Building a chemical ontology using methontology and the ontology design environment. *IEEE Intelligent Systems* 14(1), 37–46 (1999)
8. Gasevic, D., Djuric, D., Devedzic, V.: *Model Driven Architecture and Ontology Development*. Springer, Berlin (2006)
9. Devedzic, V.: Understanding Ontological Engineering. *Communications of the ACM* 45(4), 136–144 (2002)
10. de Moor, A., Leenheer, P.D., Meersman, R.: Dogma-mess: A meaning evolution support system for interorganizational ontology engineering. In: *Proc. of the 14th International Conference on Conceptual Structures (ICCS 2006)*, Aalborg, Denmark, July 17–21 (2006)
11. Kotis, K., Vouros, G.: Human-centered ontology engineering: The HCOME methodology. *Knowledge and Information Systems* 10(1), 109–131 (2006)
12. Staab, S., Studer, R., Schnurr, H., Sure, Y.: Knowledge processes and ontologies. *IEEE Intelligent Systems* 16(1), 26–34 (2001)
13. Sure, Y.: A tool-supported methodology for ontology-based knowledge management. In: *ISMIS 2002, Methodologies for Intelligent Systems* (2002)
14. Gómez-Gauchía, H., Díaz-Agudo, B., González-Calero, P.: Towards a pragmatic methodology to build lightweight ontologies: a case study. In: *Procs. of the IADIS International Conference, Applied Computing 2004*, Lisboa, Portugal (2004)
15. Gómez-Gauchía, H., Díaz-Agudo, B., González-Calero, P.: Two-layered approach to knowledge representation using conceptual maps and description logics. In: Cañas, A.J., Novak, J.D., González, F.M. (eds.) *Concept Maps: Theory, Methodology, Technology*, *Proc. of the First Int. Conference on Concept Mapping* (2004)
16. Pinto, S., Staab, S., Sure, Y., Tempich, C.: Ontoedit empowering swap: a case study in supporting distributed, loosely-controlled and evolving engineering of ontologies (diligent). In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) *ESWS 2004*. LNCS, vol. 3053. Springer, Heidelberg (2004)
17. Aschoff, F.: Knowledge mediation: A procedure for the cooperative construction of domain ontologies. Master's thesis, University of Heidelberg (2004)
18. Zhao, G.: AKEM: an ontology engineering methodology in FF POIROT. Deliverable 6.8 of FF POIROT project (2005)
19. Camarinha-Matos, L.: Collaborative networks in industry – Trends and foundations. In: *Proc. of DET 2006 - 3rd International CIRP Conference in Digital Enterprise Technology*. Springer, Heidelberg (2006)

Toward a Community Vision Driven Topical Ontology in Human Resource Management

Damien Trog¹, Stijn Christiaens¹, Gang Zhao¹, and Johanna de Laaf²

¹ Semantics Technology and Applications Laboratory (STARLab)
Department of Computer Science
Vrije Universiteit Brussel
Pleinlaan 2, B-1050 BRUSSELS 5, Belgium

{dtrog,stichris,gzhao}@vub.ac.be

² British Telecom (BT)
Global Services Learning Solutions
Offices Minerva & Mercurius
Herikerbergweg 2, 1101 CM Amsterdam Zuid-Oost, The Netherlands
johanna.delaaft@bt.com

Abstract. Today's industries require ontology engineering to be more community vision driven and ontological resources have more dimensions. This paper shows our achieved goals in the EU Prolix project. Firstly, we analyze the user requirements, including tuning Business Process Modelling efforts into ontology engineering tasks. Secondly, we design a scalable and community driven architecture for ontology development. Then, feasible ontology models are created. The task of creating ontology models depends heavily on conceptual architecture of ontology based competence analysis. We illustrate with the requirement analysis from BT (British Telecom).

1 Introduction

In this paper we discuss how we approach the architecture of ontologies in the Prolix project (FP6-IST-027905) [12]. The objective of PROLIX is to align learning with business processes in order to enable organisations to faster improve the competencies of their employees according to continuous changes of business requirements.

We extend on previous work in ontology architecture [10] aimed at improving scaleability and versatility in ontology engineering. Zhao describes four different layers in this architecture: the base ontologies (e.g., SUMO), the domain ontologies (e.g., finance, economics), topical ontologies (e.g., fraud ontology) and the application ontologies (e.g., VAT fraud prevention). On the methodological side we adopt DOGMA-MESS [1], with four different levels: meta-ontology, upper common ontology, lower common ontology and organisational ontologies. Each level is governed by a different actor (knowledge engineer, core domain expert, domain expert), which facilitates the evolution in a community-driven manner.

¹ <http://www.prolixproject.org/>

We illustrate with the requirement analysis in the BT (British Telecom) case. In this case we define the events and changes in business processes in terms of competences. This allows performance competence analysis of reality and future, management of training in terms of competences, and source trainings by competence diagnosis.

The ontology is architected into five layers: (i) Ontological Basis, (ii) Ontological Assertions, (iii) Ontological Topics, (iv) Ontological Application, and (v) Ontological Instances.

On the Ontological Topics layer, two main modules are modelled: the Workplace Individual Competency Ontology (WICO), and the Workplace Context Ontology (WCO). On the Ontological Application layer, BT training purposes, BT values (trustworthy, helpful, etc.) and BT business process in general are modelled.

2 Related Work

Ontology architecture is an area of ontology engineering which receives only little attention, despite the fact that proper (collaborative) ontology development can only be performed in a supporting structure. A similar need is present in the development and maintenance of large information systems, where various architectural approaches have been described and are being used (e.g., [5], [7], [9]).

The concept of upper ontologies is well established, and is as such also used in ontology engineering. In this approach, a high-level layer containing general, cross-domain concepts are stored and managed, meant to be used in more specific domain ontologies. Examples of such upper ontologies are SUMO [4] and DOLCE [2].

3 The BT Case

In the BT case requirements are situated in three operational processes: business process management (BPM) in fault handling in customer care, human resources management (HRM) with staff review and career path, and vocational training management (VTM).

Events cause the change in business processes and competence requirements by introducing new products or services, work flows, policies, customers and suppliers, infrastructure and tools. The competence requirements start the process in HRM by the search of competences, competence profiling to define the identity and difference, and assignment of responsibility. The process in the HRM also generates activities such as staff review and overall performance assessment.

Vocational training is targeted to job responsibilities and associated skills. Its purpose is to improve the competency of the individual to increase overall performance of the team. This requires skill oriented training materials and programs, and requirements of training for given job responsibilities associated with junior, acting and senior staff.

The underlying principle of the interaction among the three processes, HRM, BPM and VTM is centred on the concept of staff competence at work.

Its test case requirements can be summarised as follows: (i) define the events and changes in business processes in terms of competences, (ii) performance competence analysis of reality and future, (iii) manage trainings in terms of competences, (iv) source trainings by competence diagnosis.

It can be concluded in IT terms that: (i) the central resource is a vocabulary to define competences explicitly, e.g., what is “soft skills” and “customer communication”, (ii) profiles need annotating by this vocabulary, (iii) profiles need comparison in terms of this vocabulary, (iv) profiles need improvement by training managed by this vocabulary.

This vocabulary is an ontology of conceptual entities and relationships in the three processes. It will be used in data annotation, gap analysis and contents management. Ontology are used in the BT test cases for competence gap analysis and as indices to training resources. The ontology to be developed is an application ontology. The core subject of the ontology is the competency of employees at workplaces at BT. The concept of competency underlies four application domains: (i) human resources management: evaluation of staff performance, and career paths within BT; (ii) business process management: customer care process, such as fault handling, and functional roles; (iii) training: resources in view of functional roles and competency, and programs to support career development (iv) technologies: tools used, and products and services.

The creation of the ontology therefore draws from information about these four subject fields supplied by BT.

4 Ontology Architecture

The ontology architecture extends on DOGMA (Developing Ontology Grounded Methods and Applications), an ontology approach and framework not restricted to a particular representation language. One of its most characteristic features is the use of a layered architecture, in particular the possibility of having multiple views on and uses of the same stored conceptualisation. A DOGMA ontology consists of a *lexon base* layer and a *commitment layer*. This layering approach enhances the potential for re-use and allows for scalability in representing and reasoning about formal semantics [10].

The goal of the lexon base is to reach a common and agreed understanding about the ontology terminology and is thus aimed at human understanding. The *commitment layer*, with its formal constraints [3], is meant for interoperability issues between information systems, software agents and web services.

We identify five layers in the architecture of the ontology for the BT case:

1. ontological basis: a type hierarchy of generic objects;
2. ontological assertions: relations between the generic objects;
3. ontological topics: domain specific themes with a given set of assertions;
4. ontological application: business objects in terms of domain topics and ontological assertions;
5. ontological instances: a dictionary of instances or instantiations of the concepts.

4.1 Ontological Basis

The base layer is a container of essential conceptual entities, called, terms. They are type definitions of abstract or concrete objects in the four subject fields.

It also specifies how these terms are related in a generic sense. The relationship is a type of meta-information, such as subtype, supertype, instance, equivalence. It is put in the form of lexons, as any other type of relations among the terms.

The contents on the base layer constitute the basic vocabulary with which we ‘Describe’, ‘Assert’ and ‘Instantiate’ about semantics of the four domains mentioned above. For example SUMO [4] can be reused as an ontological basis.

4.2 Ontological Assertions

The ontological assertion layer consists of ontological assertions using the terms defined on the ontological base layer. The relationship between the terms is not restricted to any type. Depending on the nature of terms, the relationship can be dynamic or static; it can be concerned with actions, activities, properties, location, manner, etc. They are fact types.

Headterm	Role	Tailterm
Person	Coach	Person
Person	Anticipate	Need
Person	Acknowledge	Mistake

4.3 Ontological Topics

The topical layer specifies topics or themes relevant to the subject domain of one or a family of applications. Here a topic means a set of assertions about a domain concept. It provides means to modularise, group or package the assertions. Since topics can be structurally related, these topical modules form into a topical map for knowledge management in semantic modelling as well as how lexons are ‘located’ the domain concept map. They serve as a library of conceptual patterns or frameworks to specify business objects on the Ontological Application layer, create or extend topics, and organise topics.

It is also the layer where the other existing ontological models are interfaced or incorporated.

We recognise two main topics relevant to the BT test case vision, viz. the “workplace competency” (individual employee) and “workplace context” (organisational, process and technology).

Workplace Context Ontology. The context can be generic or specific to a family of applications and systems. Workplace Context Ontology (WCO) is concerned with the generic structure and types of the operational context of a business entity, such as BT Global Services. We propose a typology of contexts shown in Fig. 1. It is visualised in the T-Lex ontology editor [8], where the role “SupertypeOf” is interpreted as a taxonomical relationship.

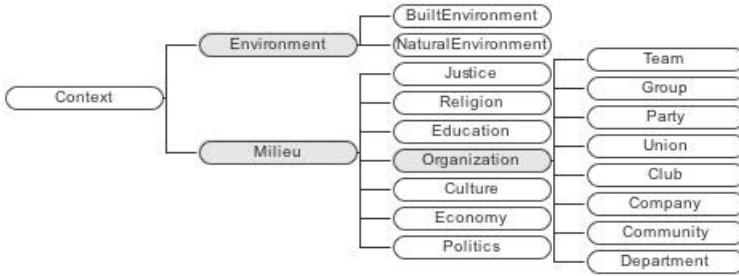


Fig. 1. Typology of context

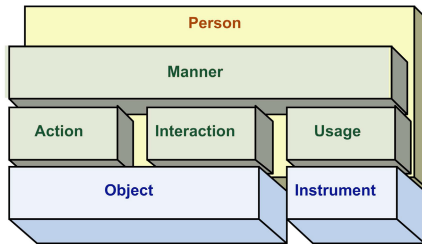


Fig. 2. The topical modules in WICO

The specific contexts or their concrete embodiment are dealt with on the layer of Ontological Instances.

International standards are part of the context, e.g., United Nations Standard Products & Services Code, North American Industry Classification System, EU Classification of Products by Activity, NACE economic sectors.

Workplace Individual Competency Ontology. The Workplace Competency Ontology is oriented towards the staff performance and its evaluation and improvement, because of the operational context of BT Global Service and its test case requirements. In consideration of this specific functional emphasis, we adopt the OECD key competence framework [6] for the overall structure of Workplace Individual Competency Ontology (WICO).

The competence framework structures the individual competences into three components: “Use tools interactively”, “Interact in heterogeneous groups”, and “Act autonomously”.

These three perspectives of competency capture well the workplace competence with respect to performance. They constitute three essential subtopics of WICO. They are concerned with use, act and interact. In addition, we add four other subtopics: *Instrument* to be used and *Manners* to act and interact, following the design pattern of ontological architecting [11].

Figure 2 shows the topical modules in WICO. It consists of ontologies about action, interaction, use of tools, object, instruments and person. It is a semantic semiotics to describe: **Persons Act or Interact on Objects by Instruments in Manners.**

Use. The topic of use pertains to the ability to use a cognitive or physical instrument. The instrument includes language, logic and mathematics, information, knowledge and tools. The key competences are: communication ability with languages, analytical ability with logic and mathematics, knowledge about the task at hand, and skill with hard- and software tools. They are defined in terms of assertions.

Group	Headterm	Role	Tailterm
C 1-A	Ability to use language, symbol and text		
	Person	Use	Language
	Person	Use	Math

Interact. The topic of Interact is concerned with the ability of team work. This includes relating, co-operating and resolving conflicts with other people. We give the ability of relating and cooperating as an example.

Group	Headterm	Role	Tailterm
C 2-A	Ability to relate well		
	Person	InteractWith	Person
	Person	Respect	Context
	Person	Empathise	Person
	Person	Manage	Emotion
C 2-B	Ability to cooperate		
	Person	Present	Idea
	Person	Accept	Idea
	Person	DebateWith	Person
	Person	Observe	Agenda
	Person	AllyWith	Person
	Person	NegotiateWith	Person
	Person	Support	Person
	Person	Lead	Person

Act. The topic of Act is concerned with the ability to function independently. Besides the abilities to use instruments appropriately, there are three key competences: analyze problems by evaluating goals and solutions with an awareness of contexts, design and implement action plans and manage projects, and assert and defend a position.

Group	Headterm	Role	Tailterm
C 3-A	Ability to act within a big picture		
	Person	Take	Decision
	Person	Take	Action
	Person	Understand	Context
	Action	Cause	Consequence

Manner. The topic of Manner groups assertions that characterise the attitudes, behaviour and activity at a workplace. It is related to work ethics, code of conduct at an institution. The following table shows examples of assertions about Manner of Person, Act and Interact.

Headterm	Role	Tailterm
Honesty	Characterise	Person
Reliability	Characterise	Person
Simplicity	Characterise	Action
Accuracy	Characterise	Interaction

4.4 Ontological Application

The layer of Ontological Application is where conceptions specific to a particular business or system are modelled using the lexons from the Ontological Assertions and Ontological Topics. Here the lexons are selected to represent particular business facts and constrained with commitments with respect to business logics.

The Ontological Abstract is a description in terms of ontological assertions and topics of anything, be it a training material, work profile, business data or process, company policy. Anything that needs knowledge management in semantic terms. Formally, it is a list of lexons. It serves as a semantic representation with respect to particular applications. It needs further constraining and instantiating.

BT’s training strategies are based on technical knowledge, soft skills and process knowledge. Below is an example of the ontological abstract in terms of competences for customer communication.

Technical Knowledge	Ontological Abstract
C-1-A	ability to use language, symbol and text
C-2-A	ability to relate well
C-2-B	ability to cooperate

4.5 Ontological Instances

The Ontological Abstract is a semantic statement about a business entity or artefact. The statement needs further grounding logically with constraints and values of instantiation. The Ontological Instances are lists of concrete values for specific instantiations of lexons. Instances can be collected according to applications from the same fields as specified in the ontological applications.

5 Ontology Application in the BT Case

Given the ontological concepts and relationships, the business objects can be described explicitly as ontological abstract. The ontological abstract is the semantic data with which any semantic-driven processing can take place.

The crucial added value of the ontology is to express the meaning explicitly either for human operation on clear and precise understanding or for automatic processing and resource management.

5.1 Competence Analysis

The competence analysis in the BT test case is to identify competence gaps at the event of a new product or service, work flow, policy, customer and supplier, or infrastructure and tools. The competence gap is between the expected competences and the existing competences. Once the gap is identified, the resources can be managed to close the gap, for example, by administration of trainings.

Given an Existing Competence Profile (ECP) and a Required Competence Profile (RCP), the process computes the competence gap, CG . $RCP - ECP = CG$. The gap analysis can be either qualitative or quantitative computation.

It breaks down into: (i) creating the ontological abstract of ECP in terms of the ontology, ECP Abstract; (ii) creating the ontological abstract of RCP in terms of the ontology, RCP Abstract; (iii) scoring ECP Abstract and RCP Abstract; (iv) computing the Competence Gap in the abstract of Competence Gap, CG Abstract, either in quantity or quality.

5.2 Knowledge Management among BPM, HRM and Training

The existence of the Workplace Competency Ontology enables the semantic link among the work flow, staff evaluation and trainings. The ontology can be used to create semantic indexes to the competence profile of an agent that performs a function in the work flow and educational purposes of training courses and materials. In other words, we can envisage a repository of training materials, events, etc. that are indexed in terms of the ontological assertions and ontological

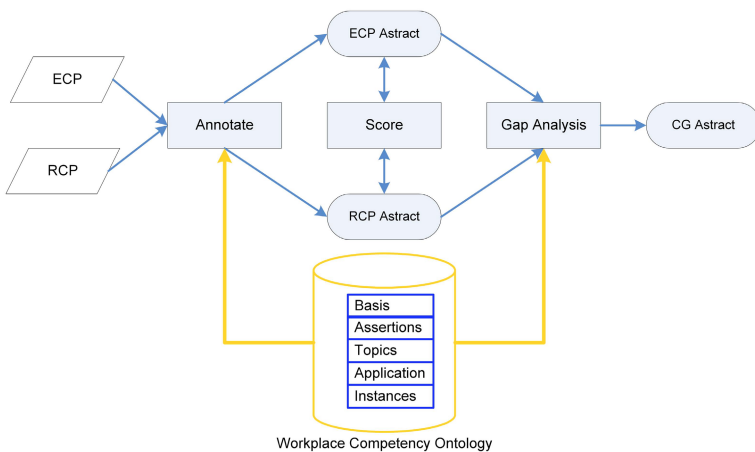


Fig. 3. Analysing the competence gap

topics. The repository can be searched and matched via an ontological abstract either about RCP or CG output by the business process of staff evaluation and work flow change.

6 Conclusions and Future Work

We have proposed an architecture of ontology in 5 layers to manage different viewpoints. Not only generic concepts and assertions are modelled but also operations are described on these layers. Two key topical ontologies are Workplace Individual Competency Ontology and Workplace Context Ontology.

The ontology modelled so far for the BT case consists of 250 lexons. It also contains 50 instances and 8 ontological abstracts about eight business concepts specific to the BT case.

We have described the conceptual architecture of the two applications of the ontology, and experimented with competence analysis.

On the issue of context ontology, it is our intention to ask three questions:

- Is it feasible to model contexts independent of competences?
- How should it interface and interact with competence ontology?
- Does it bring added value in the envisaged application?

The currently reported work has located the context ontology in the structure of the whole ontological resources. It enables us to experience initial difficulty in arbitrariness in the determination of what belongs to context and what not. Ontology is to capture different perspectives. When perspectives change, the concept of ‘context’ also changes. From semiotic systems points of view, any element in the system is the context of the other.

Acknowledgements. This research has partly been funded by the EU 6th framework program, FP6-IST-027905. The authors would like to thank Yan Tang and Christophe Debruyne for their valuable contributions.

References

1. de Moor, A., De Leenheer, P., Meersman, R.: DOGMA-MESS: A meaning evolution support system for interorganizational ontology engineering. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) ICCS 2006. LNCS (LNAI), vol. 4068, pp. 189–203. Springer, Heidelberg (2006)
2. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening ontologies with dolce. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 166–181. Springer, Heidelberg (2002)
3. Jarrar, M., Meersman, R.: Formal ontology engineering in the DOGMA approach. In: Meersman, R., Tari, Z., et al. (eds.) CoopIS 2002, DOA 2002, and ODBASE 2002. LNCS, vol. 2519, pp. 1238–1254. Springer, Heidelberg (2002)
4. Niles, I., Pease, A.: Towards a standard upper ontology. In: FOIS 2001. Proceedings of the international conference on Formal Ontology in Information Systems, pp. 2–9. ACM, New York (2001)

5. Oasis: Reference model for service oriented architecture 1.0 (2006)
6. Rychen, D.S., Salganik, L.H.: Definition and selection of key competences. In: Fourth General Assembly of the OECD Education Indicators Programme, The INES Compendium, Contributions from the INES Networks and Working Groups, OECD, Paris, France, pp. 61–73 (2000)
7. Scheer, A.W.: *Architecture of Integrated Information Systems: Foundations of Enterprise Modelling*. Springer, New York (1994)
8. Trog, D., Vereecken, J., Christiaens, S., De Leenheer, P., Meersman, R.: T-lex: A role-based ontology engineering tool. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops*. LNCS, vol. 4278. Springer, Heidelberg (2006)
9. Zachman, J.A.: A framework for information systems architecture. *IBM Syst. J.* 26(3), 276–292 (1987)
10. Zhao, G., Meersman, R.: Architecting ontology for scalability and versatility. In: Meersman, R., Tari, Z. (eds.) *OTM 2005*. LNCS, vol. 3761, pp. 1164–1605. Springer, Heidelberg (2005)
11. Zhao, G., Meersman, R.: Towards a topical ontology of fraud. In: Mizoguchi, R., Shi, Z.-Z., Giunchiglia, F. (eds.) *ASWC 2006*. LNCS, vol. 4185, pp. 566–572. Springer, Heidelberg (2006)
12. Zhao, G., Stam, J., Delaaf, J., Debruyne, C., Christiaens, S., Tang, Y.: Test bed vision analysis and first version ontologies: BT case. *Prolix Deliverable 3.4*, STARLab (2008)

Automatic Profiling System for Ranking Candidates Answers in Human Resources

Rémy Kessler¹, Nicolas Béchet², Mathieu Roche², Marc El-Bèze¹,
and Juan Manuel Torres-Moreno^{1,3}

¹ LIA, BP 1228 F-84911 Avignon cedex 9, France

² LIRMM - UMR 5506, CNRS - Univ. Montpellier 2 - France

³ École Polytechnique de Montréal - Département de génie informatique
CP 6079 Succ. Centre Ville H3C 3A7, Montréal (Québec), Canada
{remy.kessler, juan-manuel.torres, marc.elbeze}@univ-avignon.fr
{nicolas.bechet, mathieu.roche}@lirmm.fr

Abstract. The exponential growth of Internet allowed the development of a market of online job search sites. This work aims at presenting the E-Gen system (Automatic Job Offer Processing system for Human Resources). E-Gen will implement several complex tasks: an analysis and categorization of jobs offers which are unstructured text documents (e-mails of job offers possibly with an attached document), an analysis and a relevance ranking of the candidate answers. We present a strategy to resolve the last task: After a process of filtering and lemmatisation, we use vectorial representation and different similarity measures. The quality of ranking obtained is evaluated using ROC curves.

1 Introduction

The exponential growth of Internet allowed the development of an online job-search sites market [1,2]. The answers of candidates represent a lot of information that can not be managed efficiently by companies [4,5]. It is therefore indispensable to process this information by an automatic or assisted way. Thus, we develop the E-Gen system to resolve this problem. It will be composed of three main modules:

1. A module of information extraction from a corpus of e-mails of job offers.
2. A module to analyse the candidate answers (splitting e-mails into Cover Letter and Curriculum Vitae).
3. A module to analyse and compute a relevance ranking of the candidate answers.

Our previous works present the first module [7], the identification of different parts of a job offer and the second module [8] which analyses the contents of a candidate e-mail with Support Vector Machine [9] and n -gramms approach. We present in this paper a strategy to resolve the last module. The large number of candidates answers for a job generates a hardly process of reading for the

recruiting consultant. In order to facilitate this task, we propose a system capable of providing an initial evaluation of candidates answers according to various criteria. We will show which document (Curriculum Vitae or Cover Letter) contains the most relevant information and the location of this relevant information in each document. We use different similarity measures between a job offer and candidates answers to rank them. Section 2 shows a general system overview. In section 3 we describe the pre-processing task and the different measures used to rank the candidates answers. In section 4 we present statistics about textual corpus, examples (job offer, Cover Letter and Curriculum Vitae), experimental protocol. Results and discussions are presented in 5.

2 System Overview

The main activity of Aktor¹ Interactive is the processing of job offers on the Internet. As the Internet proposes new ways to the employment market, Aktor modifies its procedures to integrate a system which answers as fast and judiciously as possible to this processing. An e-mail-box receives messages containing the offer. After language identification, E-Gen parses the e-mail, splits the offer in segments, and retrieves relevant information (contract, salary, location, etc.) to put on line job offer. During the publication of jobs offer, Aktor generates an e-mail address for applying to the job.

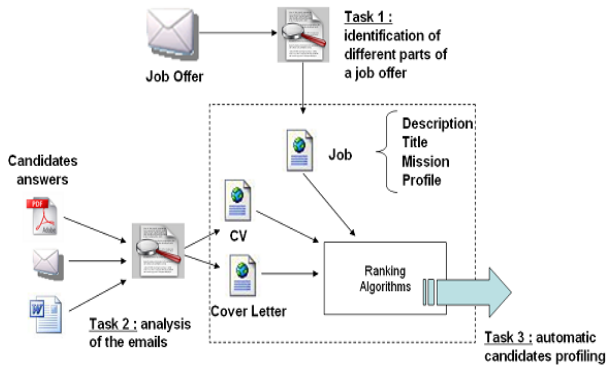


Fig. 1. System overview

Each e-mail is so redirected to a Human Resources software, Gestmax² to be read by a recruiting consultant. At this step, E-Gen analyses the candidate's answers to identify each part of the candidacy and extract text from e-mail and enclosed files (resp. wvWare³ and pdftotext⁴ for MS-Word and PDF documents).

¹ <http://www.aktor.fr>

² <http://www.gestmax.fr>

³ <http://wvware.sourceforge.net>

⁴ http://www.bluem.net/downloads/pdftotext_en

At this last step, E-Gen analyses candidate's answer for ranking candidates for the mission. The system architecture of E-Gen is represented in figure 11.

3 Ranking Algorithms

3.1 Corpus Pre-processing

A pre-processing task of the corpus is performed to obtain a suitable representation in Vector Space Model (VSM). In order to avoid introduction of noise in the model, the following items are deleted: Verbs and stop words (to be, to have, to need,...), common expressions with a stoplist⁵ (for example, that is, each of,...), numbers (in numeric and/or textual format), and symbols such as \$, #, *, etc. Lemmatisation processing is also performed to obtain an important reduction of the lexicon. It consists in finding the root of verbs and transform plural and/or feminine words to masculine singular form⁶. This process allows to decrease the curse of dimensionality [11] which raises severe problems of representation of the huge dimensions [12]. Other mechanisms of reduction of the lexicon are also used: compounds words are identified by a dictionary, then transformed into a unique term. All these processes allow us to represent the collection of documents through the bag-of-words paradigm (a matrix of frequencies/absences of segment texts (rows) and a vocabulary of terms (columns)).

3.2 Measures of Similarity

We use a number of similarity measures to determine which is most effective. Firstly we use Enertex similarity. Textual energy measure was successfully used in different NLP (Natural Language Processing) tasks as automatic summarization and topic segmentation [13]. Based on energy of the Ising magnetic model, it considers a document of n terms as a chain of n binary units called spins. Up spins represents present words and down spins the absents one. In our model, we are particularly interested in textual energy between the job offer and each candidate as:

$$J^{i,j} = \sum_{\mu=1}^P s_{\mu}^i s_{\mu}^j \quad (1)$$

$$E_{\mu,\nu} = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N s_{\mu}^i J^{i,j} s_{\nu}^j \quad (2)$$

With s_{μ}^i = term i of job offer, s_{ν}^j = term j of candidate, J_{ij} interaction of terms i , and j in the P candidature's documents.

The next measure is the Needleman-Wunsch algorithm [14], commonly used in bioinformatics to align sequences of nucleotides. We consider vector of job offer j and vector of candidate answer d as sequence of terms and we compute the best

⁵ <http://sites.univ-provence.fr/~veronis/donnees/index.html>

⁶ So we conflate terms *sing*, *sang*, *sung*, *will sing* and possibly *singer* into *sing*.

score $H(j, d)$ between the two sequences with the algorithm. We also tested different similarity measures as defined in [15]: *cosine* (3), which allows us to calculate cosinus of the angle between job offer and each candidate answer, the Minkowski distances (4) ($p = 1$ for Manhattan, $p = 2$ for euclid) and overlap (5).

$$sim_{cosine}(j, d) = \frac{j_i \cdot d_i}{\sqrt{\sum_{i=1}^n |j_i|^2 \cdot \sum_{i=1}^n |d_i|^2}} \tag{3}$$

$$sim_{Minkowski}(j, d) = \frac{1}{1 + (\sum_{i=1}^n |j_i - d_i|^p)^{\frac{1}{p}}} \tag{4}$$

$$sim_{Overlap}(j, d) = \frac{j_i \cdot d_i}{Min(\sum_{i=1}^n |j_i|^2, \sum_{i=1}^n |d_i|^2)} \tag{5}$$

With j job offer, d a candidate answer, i a term. The last measure used is Okabis[16]. Based on okapi formula, a measure often used in Information Retrieval :

$$Okabis(j, d) = \sum_{w \in d \cap j} \frac{TF_{w,d}}{TF_{w,d} + \frac{\sqrt{|d|}}{M_S}} \tag{6}$$

With j job offer, d a candidate answer, w a term, $TF_{w,d}$ occurrence of w in S , N total numbers of candidates and M_S their average size. To combine these measures, we use an Algorithm Decision (AD) [17], which will weigh the values obtained by each measure of similarity. Two averages λ are calculated: The positive tendency, (that is $\lambda > 0.5$), and the negative tendency, for ($\lambda < 0.5$) and then a decision is calculated.

4 Experiments

We have selected a data subset from Aktor’s database. This corpus, called *Reference corpus*, is a set of job offers with various thematics (jobs in accountancy, business enterprise, computer science, etc.) associated with their candidates. Each candidates is tagged **positive** or **negative**. A **positive** value correspond to a potential candidate for a given job by the recruiting consultant. A **negative** value is for a irrelevant candidate for the job (decision of the company). Initially we concentrated our study on french job offers because the french market represents the main Aktor’s activity. Table 1 shows a few statistics about the *Reference corpus*.

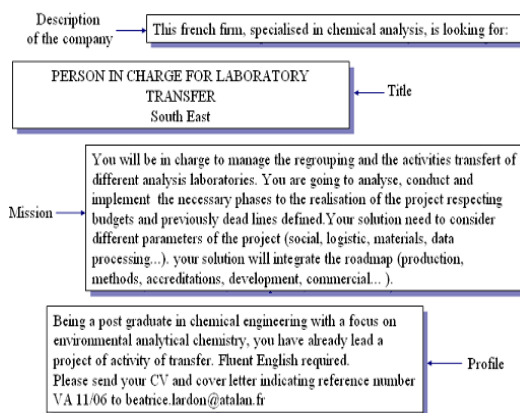
4.1 Example of Job Offer

Each job offer is separated in four parts, as defined in [7] :

1. 'Description_of_the_company': Brief abstract of the company that recruits.
2. 'Title': job title.
3. 'Mission': a short job description.

Table 1. Corpus statistics

Total number of jobs offers	25
Number of jobs offers with less 10 candidates	2
Number of jobs offers with more 10 candidates	8
Number of jobs offers with more 50 candidates	6
Number of jobs offers with more 100 candidates	9
Total Number of candidates	2916
Number of candidates with tagging positive	220
Number of candidates with tagging negative	2696

**Fig. 2.** Job offer segmented

- 'Profile': required skills and knowledge for the position. Contacts are generally included in this part.

A job offer example with its segmentation is presented in figure 2, translated in English. The content of the job offer is free but we find a rather similar presentation and vocabulary according to every part. This segmentation is used for ranking candidates as we shall see it afterwards in section 5.

4.2 Example of Candidature

Figure 3 is a example of French Curriculum Vitae with a translation in English and figure 4, a French Cover Letter with a translation in English (documents were previously automatic anonymized). A first analysis of Curriculum Vitae and Cover Letter shows that the documents are very different. Cover Letter is a complete text, containing references to the job offer while CV summarizes career of candidates. The content of CV are free but we find a rather similar presentation and vocabulary according to each block ("Professional experience", "Educational background", "Personal interests", etc.) and some relevant collocations [10] as "assistant director", "Computer skills", "Driver licence", etc.

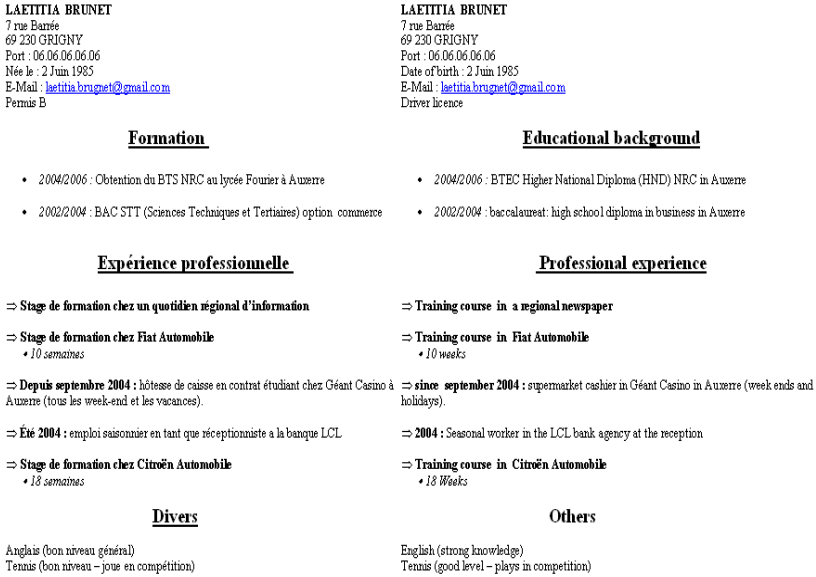


Fig. 3. Example of Curriculum Vitae in french on left and in english on the right

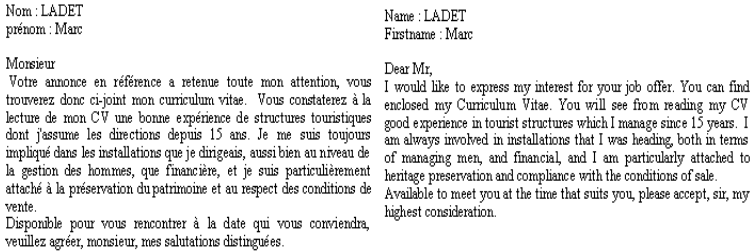


Fig. 4. Example of Cover Letter in french on left and in english on the right

4.3 Experimental Protocol

We propose to measure the similarity between a job offer and the candidates. We have 25 job offers associated to at least 4 candidates. We represented textual information in Vector Space Model [18], then several similarity measures are applied. These measures ranks the candidate answers by computing a similarity between a job offer and their associated candidates answers. We apply several similarity measures presented in section 3.2: Enertex, cosine, Minkowski, Manhattan, Needleman-Wunsch, and Overlap. Finally, we use Decision Algorithm which combines the previews ones. We use the ROC Curves to evaluate the quality of obtained ranking with tagging defined in [4]. Initially the ROC curves in [19], come from the field of signal processing. ROC curves are often used in medicine to evaluate the validity of diagnostic tests. The ROC curves show in

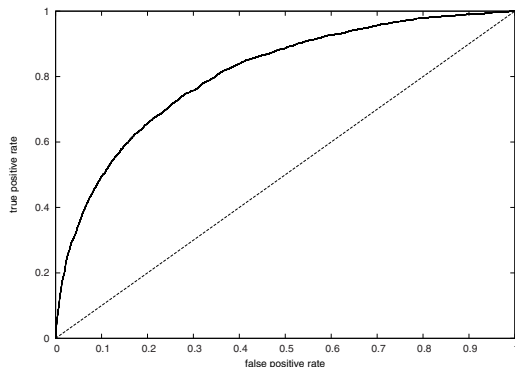


Fig. 5. Example of ROC Curve

X-axis the rate of false positive (in our case, rate of irrelevant candidate answers) and in Y-axis the rate of true positive (rate of relevant candidate answers). The surface under the ROC curve (AUC - *Area Under the Curve*), can be interpreted as the effectiveness of a measurement of interest. The criterion relating to the surface under the curve is equivalent to the statistical test of Wilcoxon-Mann-Whitney [20]. In the case of candidate answers ranking, a perfect ROC curve corresponds to obtain all relevant candidate answers at the beginning of the list and all irrelevant at the end. This situation corresponds to $AUC = 1$. The diagonal line corresponds to the performance of a random system, progress of the rate of true positive being accompanied by an equivalent degradation of the rate of false positive. This situation corresponds to $AUC = 0.5$. The figure 5 is an example of a ROC Curve with in diagonal a random system distribution. If the candidate answers are ranked by decreasing interest (*i.e.* all relevant candidate answers are after the irrelevant ones) then $AUC = 0$. An effective measurement of interest to order candidate answers consists in obtaining an AUC the highest possible value. This is strictly equivalent to minimizing the sum of the rank of the positive examples. The advantage of ROC curves comes from its resistance to imbalance (for example, an imbalance in number of positive and negative examples). For each job offer, we evaluated the quality of ranking obtained by our method. We computed AUC averages over 25 job offers.

5 Results

First we propose to study the structure of data (job offer and candidate answer). A job offer is composed by a description, a title, a mission, and a profile. We use two combines of a job offer content:

- conserving only Title, Mission, Profile (called TMP)
- conserving all information of a job offer (called DTMP)

A candidate answer is composed by a CV and a CL.

Table 2. AUC obtained with our different filtering

	AUC	Enerterx	Cosine	Minkowski	Manhattan	NW	Overlap	Okapi	Decision
DTMP	CL	0,524	0,567	0,561	0,591	0,481	0,573	0,521	0,596
	CV	0,524	0,604	0,510	0,503	0,532	0,543	0,541	0,562
	CL+CV	0,523	0,621	0,539	0,532	0,509	0,522	0,523	0,571
TMP	CL	0,524	0,560	0,559	0,580	0,473	0,562	0,513	0,591
	CV	0,523	0,622	0,508	0,501	0,544	0,538	0,542	0,561
	CL+CV	0,523	0,642	0,538	0,528	0,526	0,531	0,532	0,592

Table 3. AUC obtained with different parts of CV and CL

AUC	Enerterx	Cosine	Minkowski	Manhattan	NW	Overlap	Okapi	Decision
CV_1/3	0,525	0,589	0,497	0,505	0,533	0,539	0,569	0,579
CV_2/3	0,524	0,600	0,524	0,520	0,515	0,577	0,560	0,580
CV_3/3	0,526	0,526	0,497	0,503	0,510	0,479	0,506	0,501
CL_1/3	0,527	0,573	0,561	0,588	0,480	0,571	0,528	0,580
CL_2/3	0,533	0,565	0,570	0,578	0,481	0,578	0,543	0,570
CL_3/3	0,516	0,447	0,528	0,538	0,416	0,446	0,439	0,470

Table 2 presents the AUC results by studying the impact of the CV and CL in candidate answers with DTMP and TMP. We use the different similarity measures presented in 4.3 to compute ranking. The best AUC is obtained with the cosine measure, using the CV and CL of candidate answers and the TMP job offer. The cosine measure gives best results for all approaches, except by considering CL only (with DTMP and TMP), which is better with the Decision similarity measure. We assume that the results obtained by the Decision Algorithm are noisy by the poor performance of certain measures (Overlap or Needleman-Wunsch). The Enerterx measure has strangely very similar results for all kind of data, we work to determine the reasons of these results. We observe that using CV is more relevant than CL. These results confirm our intuition, the CV is the main document and contains the most important information of a candidature. Globally, AUC scores are not very relevant. We can explain this fact by the nature of data and the quality of the human expertise. Finally, we propose to split CV and CL in three parts to identify the parts containing relevant knowledge. Table 3 presents AUC of splitted CV and CL. Figure 6 presents graphical results of table 3 by using best similarity measures (Cosine, Minkowski, Manhattan, Overlap, and Decision).

We obtain irrelevant scores with the last part of CV and CL. We conclude that the relevant knowledge to determine a candidate adapted to an offer, is contained in the 1/3 and 2/3 of the CV and CL. Actually, the last part of CV is generally "Hobbies" or "Other" which is rarely a crucial information. In the same way, the last part of CL is generally as "Yours faithfully", "Yours sincerely", "best regards", which are irrelevant informations. We are currently working a finer segmentation for each document.

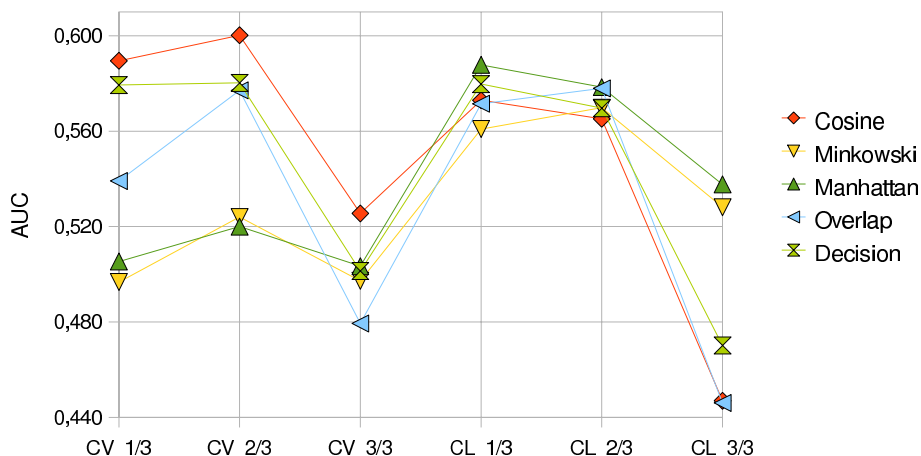


Fig. 6. Best similarity measures comparison with parts of CV and CL

6 Conclusion and Future Work

Processing job information is a difficult task because the information flow is still strongly unstructured. In this paper we show the ranking module, the last component of E-Gen, a modular system to analyse the candidate answers. We tested different measures of similarity and different segmentation of the job offer, Curriculum Vitae and Cover Letter. Cosine similarity seems offer the best results in this task (probably the IR representation favours this measure), but we prefer to combine several measures with a more stable AD. The first results obtained are interesting and we are considering best segmentation to improve performance. We are currently testing our system by using a part-of-speech tagger to improve performance of preprocessing. The first and second module of E-Gen are currently in test on Actor's server and allows a considerable saving of time in the daily treatment of job offers. E-Gen is database multilingual, independent and portable, because it is a modular system with e-mail in input and XML documents as output. We are also setting up a system for evaluating Curriculum Vitae on employment portal *jobmanager*⁷ allowing internauts to find the jobs offers that better match with their profile. In our future work, we propose to add domain knowledge as Human Resources ontologies [19] to improve the quality of the results.

References

1. Bizer, C., Heese, R., Mochol, M., Oldakowski, R., Tolksdorf, R., Eckstein, R.: The impact of semantic web technologies on job recruitment processes. In: International Conference Wirtschaftsinformatik (WI 2005), Bamberg, Germany (2005)
2. Rafter, R., Bradley, K., Smyt, B.: Automated Collaborative Filtering Applications for Online Recruitment Services, 363–368 (2000)

⁷ <http://www.jobmanager.fr>

3. Rafter, R., Smyth, B.: Passive Profiling from Server Logs in an Online Recruitment Environment
4. Bourse, M., Leclère, M., Morin, E., Trichet, F.: Human resource management and semantic web technologies. In: Proceedings, 1st International Conference on Information & Communication Technologies: from Theory to Applications, ICTTA (2004)
5. Morin, E., Leclère, M., Trichet, F.: The semantic web in e-recruitment. In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) ESWS 2004, vol. 3053, Springer, Heidelberg (2004)
6. Rafter, R., Smyth, B., Bradley, K.: Inferring Relevance Feedback from Server Logs: A Case Study in Online Recruitment
7. Kessler, R., Torres-Moreno, J.M., El-Bèze, M.: E-Gen: Automatic Job Offer Processing system for Human Resources. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, Springer, Heidelberg (2007)
8. Kessler, R., Torres-Moreno, J.M., El-Bèze, M.: E-Gen: Profilage automatique de candidatures. In: Traitement Automatique de la Langue Naturelle (TALN 2008), Avignon, France (2008)
9. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)
10. Roche, M., Prince, V.: Evaluation et détermination de la pertinence pour des syntagmes candidats à la collocation. In: JADT 2008, pp. 1009–1020 (2008)
11. Bellman, R.: Adaptive Control Processes. Princeton University Press, Princeton (1961)
12. Manning, D., Schütze, H.: Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge (2002)
13. Fernandez, S., da Cunha, I., Velázquez, P., Jorge Vivaldi, M., SanJuan, E., TorresMoreno, J.M.: Textual Energy of Associative Memories: performants applications of ENERTEX algorithm in text summarization and topic segmentation. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, Springer, Heidelberg (2007)
14. Needleman, S., Wunsch, C.: A general method applicable to the search for similarities in the amino acid sequences of proteins or nucleotides (1970)
15. Bernstein, A., Kaufmann, E., Kiefer, C., Bürki, C.: Simpack: A generic java library for similarity measures in ontologies. Technical report, University of Zurich, Department of Informatics, Winterthurerstrasse 190, 8057 Zurich, Switzerland (August 2005)
16. Patrice Bellot, M.E.B.: Classification et segmentation de textes par arbres de décision. In: Technique et Science Informatiques (TSI), Hermès, vol. 20 (2001)
17. Boudin, F., Moreno, J.M.T.: Neo-cortex: A performant user-oriented multi-document summarization system. In: CICLing, pp. 551–562 (2007)
18. Salton, G.: Developments in automatic text retrieval. *Science* 253, 974–979 (1991)
19. Ferri, C., Flach, P., Hernandez-Orallo, J.: Learning decision trees using the area under the ROC curve. In: Proceedings of ICML 2002, pp. 139–146 (2002)
20. Yan, L., Dodier, R., Mozer, M., Wolniewicz, R.: Optimizing classifier performance via an approximation to the Wilcoxon-Mann-Whitney statistic. In: Proceedings of ICML 2003, pp. 848–855 (2003)
21. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: International Conference on New Methods in Language Processing, Manchester, UK, unknown (1994)

Semantic Expansion of Service Descriptions

Christian Sánchez and Leonid Sheremetov

Mexican Petroleum Institute, Mexico.
{chsanche, sher}@imp.mx

Abstract. Recently, the market of information technologies has witnessed the explosive interest in Web services (WS), which provide several critical characteristics for development of enterprise information systems. The WS discovery, aiming in finding services meeting user's request, is an open problem yet since most of the current works both centered in syntactic (like UDDI) and semantic service descriptions, focus on finding a service with an exact match, which is not always possible. This paper presents two algorithms inspired in the query expansion of Web search engines, which expand client service descriptions (specified in OWL-S) to similar service descriptions, extending the matching types to exact, leftover and missing information. These algorithms are further used for dynamic service composition. The proposed model works with the current syntactic UDDI systems. Described algorithms are illustrated by example. The characteristics of the algorithms are studied grounded in analysis of WS and ontologies to calculate the number of service descriptions generated by the algorithm, according to a user request.

Keywords: Service Description, Semantic Similarity, Query Expansion.

1 Introduction

Recent years have witnessed an impressive rise of interest in research topics related to semantic service discovery and dynamic service composition (DSC) both in the academic communities and industrial sector. DSC permits to integrate on the fly "plug-compatible" interacting software components or remotely execute computationally hard tasks by handheld devices aiming in distributed, loosely coupled, flexible and heterogeneous software systems thus reducing the costs of enterprise applications at the same time increasing their capabilities [1]. To enable DSC, semantic interoperability of service interfaces should be ensured, grounded in a formal model that permits to describe services, identify similarities between them and finally to compose complex services. There exist several proposals to combine syntactic (UDDI) and semantic (OWL-S) properties of services [2]. On the one hand, they look for improving existing syntactic discovery techniques, on the other – for semantically expanding queries for services. This expansion may be achieved by different methods: generation of natural language service descriptions for specific domains to select the most appropriate service in the case of ambiguities [3], generation of queries using combinations of parameter's synonyms [4], etc.

Most of these approaches consider only direct matching among inputs-outputs-preconditions-effects (IOPE) or require the development of semantic UDDI [2]. The

former approach reduces significantly the set of potentially viable services, while the latter results in semantic extensions of the queries of order of several millions. In [5] a formal model of service matching with incomplete information with fuzzy definition of similarity degrees was described. This model addresses the problems of similarity checks between service descriptions for service ranking and the selection of the most similar service that satisfies the request. In this paper, an approach to the discovery of similar services with incomplete information based on the expansions of client's OWL-S service descriptions is proposed. It is inspired in query extension techniques of Web search engines where the ontology technology is effectively used [6]. The goals of this paper are i) to develop and analyze the algorithms for the expansion of service descriptions and ii) to show how they can be used for DSC.

The rest of the paper is organized as follows. In the following section the basic mechanisms of UDDI are introduced. In section 3, different types of matching are defined. In Section 4 two algorithms for the expansion of service descriptions are developed. In Section 5 a mechanism for DSC using the expansion of the service description in the case of missing information is described. In Section 6 experimental results are analyzed. Finally, the concluding remarks summarize the results obtained in this paper.

2 Syntactic and Semantic Service Matching

A UDDI (*Universal Description, Discovery and Integration*) is a service permitting register, delete and search services described in an XML-based language, like WSDL or OWL-S language. An OWL-S description of a service profile (information, which the service requires and that can provide) is represented by a 4-tuple $s_i=(I_i, O_i, P_i, E_i)$, where I_i and O_i represent respectively inputs (I) required and outputs (O) offered by a service, P_i are preconditions that should be kept for the service functioning, and E_i are effects of the service execution. Each service s_i pertains to a category ST_i .

Client's request s_C is generated in order to find a service offered by a service provider s_ϕ registered in UDDI. It is said that the UDDI finds a service s_ϕ satisfying the equality conditions, $s_\phi = s_C$, if the inputs, outputs and services' categories are exactly the same, $I_C = I_\phi$, $O_C = O_\phi$ and $ST_C = ST_\phi$. Nevertheless, for many reasons it is not always possible, e.g. there is no such service, the service is not available at the moment or there is an ambiguity in the service description.

As mentioned before, to deal with this problem, semantically similar descriptions should be generated in order to find a service most similar to the requested one. These descriptions can be generated by the UDDI itself (e.g. semantic UDDI). The main problems of such approach are: i) they are incompatible with traditional and widely used in applications syntactic UDDI and ii) the number of generated semantically similar descriptions can be very large. On the contrary, the proposed approach explores traditional UDDI (Fig. 1). It makes use of a broker agent (or agents) that generates expanded service descriptions and then sends them to the UDDI. On the way back, this agent receives service descriptions from the UDDI for filtering and ranking based on the degrees of semantic similarity. Nevertheless, as shown below for the case of the step-by-step algorithm (Section 4.2), this task can be even omitted since the preliminary filtering occurs while the descriptions are generated.

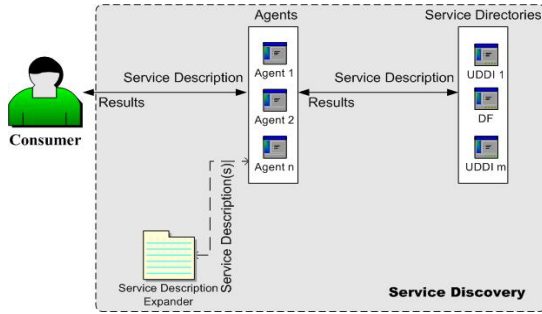


Fig. 1. Proposed approach to service discovery based on expanded service descriptions

3 Types of Service Matching

In this paper, three different service matching types are proposed. Along with the exact matching, leftover and missing types are defined enabling to find those services that match (probably to some extent) the desired inputs and outputs. Comparing the requested and offered services the following types of sets can be defined: $\bar{I} = I_C - I_\Phi$, $\bar{O} = O_\Phi - O_C$, $\bar{I} = I_\Phi - I_C$ and $\bar{O} = O_C - O_\Phi$ representing leftover inputs and outputs, and missing inputs and outputs, respectively. Possible matching types are illustrated in Table 1.

Suppose the task is to obtain the information from the data logs of a petroleum well from a certain Mexican reservoir. A user queries a service *IMPLogWebService*, requiring the well’s identifier and the date (preconditions and effects are not considered since they could form another selection criterion):

IMPLogWebService(Wellbore wellbore_input, Date date_input, Drilling_Log dl_output, Mineral_Composition mc_output)

Several queries correspond to the defined matching types and the user’s query:

- a,b) **IMPLogWebService**(Wellbore wellbore_input, Date date_input, Drilling_Log dl_output, Mineral_Composition mc_output)
- c) **IMPLogWebService**(Wellbore wellbore_input, Drilling_Log dl_output, Mineral_Composition mc_output)
- d) **IMPLogWebService**(Wellbore wellbore_input, Date date_input, Drilling_Log dl_output, Mineral_Composition mc_output, Country country_output)
- e) **IMPLogWebService**(Wellbore wellbore_input, Drilling_Log dl_output, Mineral_Composition mc_output, Country country_output)
- f) **IMPLogWebService**(Wellbore wellbore_input, Date date_input, Drilling_Log dl_output)
- g) **IMPLogWebService**(Wellbore wellbore_input, Date date_input, Hour hour_input, Drilling_Log dl_output, Mineral_Composition mc_output)
- h) **IMPLogWebService**(Wellbore wellbore_input, Date date_input, Hour hour_input, Drilling_Log dl_output)

Table 1. Types of Service Matching

Type of Matching	Description
a) Exact Service Matching	$I_c = I_\phi \ \& \ O_c = O_\phi \ \& \ ST_c = ST_\phi \ \& \ \dot{I} = \ddot{O} = \dot{I} = \ddot{O} = \emptyset$
b) Exact I&O	$I_\phi \subseteq I_c \ \& \ O_c \subseteq O_\phi \ \& \ \dot{I} = \ddot{O} = \dot{I} = \ddot{O} = \emptyset$
c) Leftover Inputs	$I_\phi \subseteq I_c \ \& \ \dot{I} \neq \emptyset \ \& \ \ddot{O} = \emptyset$
d) Leftover Outputs	$O_c \subseteq O_\phi \ \& \ \dot{I} = \emptyset \ \& \ \ddot{O} \neq \emptyset$
e) Leftover I&O	$I_\phi \subseteq I_c \ \& \ O_c \subseteq O_\phi \ \& \ \dot{I} \neq \emptyset \ \& \ \ddot{O} \neq \emptyset$
f) Missing Outputs	$O_\phi \subseteq O_c \ \& \ \dot{I} = \emptyset \ \& \ \ddot{O} \neq \emptyset$
g) Missing Inputs	$I_c \subseteq I_\phi \ \& \ \dot{I} \neq \emptyset \ \& \ \ddot{O} = \emptyset$
h) Missing I&O	$I_c \subseteq I_\phi \ \& \ O_\phi \subseteq O_c \ \& \ \dot{I} \neq \emptyset \ \& \ \ddot{O} \neq \emptyset$

The preference order for the matching types is defined as follows: *Exact* > *Exact Input&Output* > *Leftover (Inputs and/or Outputs)* > *Missing (Outputs or Inputs)* > *Missing Inputs&Outputs* > *Fail*. The concept of exact matching between inputs and outputs is extended below to their semantic similarity.

3.1 Semantic Similarity of Service Inputs and Outputs

In order to be able to define the similarity between services it is necessary to define the similarity of the elements that describe the service, i.e. inputs, outputs and the category they belong to. Since the descriptions of services are based on OWL-S, semantic similarity of these elements is defined using *Equivalence* and *Derivation* from Description Logic (DL) [7]. While comparing the input and output sets, these should be compared element by element for each set (e.g. $i_{c_1} = i_{\phi_1}$ or $o_{c_1} = o_{\phi_1}$). It is said that all the set elements are equal (e.g. $I_c = I_\phi$ and $O_c = O_\phi$), if among all of them one of the following two relations defined in [7] can be defined (*Equivalent* \equiv , *Subsumed* \supseteq). Each relation represents a type of matching according to the DL. Each input or output is represented as an instance of an OWL concept from certain ontology. Matching between these concepts pertains to one of the following types: i) *Equivalent Match* or ii) *Subsumed Match*.

Let s_c be a requested service with inputs I_c and outputs O_c . Being i_{c_n} one of the inputs of the service s_c , where $i_{c_n} \in I_c$ and $1 \leq n \leq |I_c|$, it is said that an input i_{ϕ_n} matches the requested input if $i_{c_n} \equiv i_{\phi_n}$ or $i_{\phi_n} \supseteq i_{c_n}$. Matching outputs are defined in similar fashion $o_{c_m} \equiv o_{\phi_m}$ and $o_{c_m} \supseteq o_{\phi_m}$, where $o_{c_m} \in O_c$ and $1 \leq m \leq |O_c|$.

3.2 Measuring Similarity Degrees

The similarity of the services is measured by means of their parameters. The distance between the inputs and outputs is measured as follows. An input i_{c_n} has the zero distance to the input i_{ϕ_m} , if both are described by the same concept. On the other

hand, i_{C_m} has a distance -1 to the input i_{Φ_m} if i_{Φ_m} is a parent concept for i_{C_m} . An output o_{C_n} has the zero distance to the output o_{Φ_n} , if both are described by the same concept. On the other hand, o_{C_n} has a distance equal to 1 to the output o_{Φ_n} , if this output is a child concept for o_{C_n} . In the same way the distances are calculated for any found ancestor's or predecessor's level (upcasting and downcasting for inputs and outputs respectively). To calculate the distance between the client's and provider's service descriptions, respective distances for each input and output are calculated and then the absolute values of these distances are summed.

To illustrate the above, let us consider Petroleum Wells related concepts (Table 2) that represent the service parameters and their ancestors (for inputs) and descendants (for outputs). Considering that the input parameter *Date* has no parents as well as the output *Drilling_Log* has no children, below is an example of a service description semantically similar to that requested by a client:

i) **IMPLogWebService**(*DrillHole dh_input, Date date_input, Drilling_Log dl_output, Lithology litho_output*)

Table 2. Parents and children for input and output parameters

Thing			
Hole			
BoreHole			
DrillHole	Thing		
(Wellbore,	Date,	Drilling_Log,	Mineral Composition)
		Nothing	Lithology
			Nothing

Service description *i* is similar semantically to the initial query because, if the client counts on a *Wellbore* concept, he can do cast towards *DrillHole* concept. On the other hand, a client is looking for *Mineral_Composition* and obtains *Lithology* that is a more specific concept. The subsumed match (case *i*) would generate (-1,0,0,1) or a similarity degree of 2.

4 Expanded Service Description

In the case there is no service with exact match with the one required by the client, similar service descriptions are to be generated. There are two options to expand the descriptions. First, generate all the descriptions of similar services, look for those services and then evaluate, rank and select the most similar ones. This method is called an extensive expansion. Second, generate the queries according to the similarity degree, level by level, starting from the most similar to the less similar ones. In case there are services found, they can be selected directly without additional evaluation and ranking. This method is called step-by-step expansion.

4.1 Extensive Generation of Service Descriptions

The equation (1) determines the number of similar service descriptions # q given a service description expansion in inputs q_e and outputs q_s .

$$\#q = q_e * q_s, \text{ where } q_e = \begin{cases} \prod_{j=1}^{|I|} (a_{i_j} + 1) & \text{if } 1 \leq |I| \\ 0 & \text{otherwise} \end{cases} \text{ and } q_s = \begin{cases} \prod_{k=1}^{|O|} (d_{o_k} + 1) & \text{if } 1 \leq |O| \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Each a_{i_j} represents a number of ancestors for each input i_j , where $i_j \in I$, and each d_{o_k} is the number of descendents for each output o_k , where $o_k \in O$. The algorithm for extensive service generation is illustrated in Fig. 2.

- 1: Suppose there is a service s_c with its respective j inputs I_c and k outputs O_c
- 2: If $k > 0$, then
- 3: **REPEAT** for each output k
- 4: Obtain all children m for o_k using an API of an ontology tool
- 5: If $m = 0$, then GOTO 10
- 6: **REPEAT** for each output's child m
- 7: Generate a copy of the service description s_c' substituting the output o_k by a child m
- 8: **END REPEAT m**
- 9: **END REPEAT k**
- 10: **REPEAT** for each input $0 < r < j$
- 11: Obtain all parents l for i_r .
- 12: If $l = 0$, then **EXIT**
- 13: **REPEAT** for each input's parent l
- 14: Generate a copy of the service description s_c'' substituting the input i_r by a parent l
- 15: **END REPEAT l**
- 16: **END REPEAT r**

Fig. 2. Algorithm 1: extensive generation of service descriptions

As it can be seen, at the first step if o_k has m children, new m queries are generated. At the second step, for n children of the second output, $(m+1)+[(m+1)*n]$ new service descriptions are obtained and so on. On the other hand, if each input i_r has l parents, $f+(f*1)$ new queries will be generated, where f is the number of queries generated for the outputs.

For the above example, the parents and children of all parameters from the inputs and outputs for the requested service description are illustrated in Table 2. Since *Date* and *Drilling_Log* have no parents and children respectively, additional concepts should be considered: *Thing* as a parent for inputs and *Nothing* as a child for the outputs. These concepts are used to specify the descriptions with missing inputs and outputs (see case 1b below where *Nothing* is omitted). Since *Lithology* is the only child of *Mineral_Composition*, and the input *Wellbore* has 3 parents, according to the equation (1) for this example $[(3+1)*(0+1)]*[(0+1)*(1+1)] = [4]*[2] = 8$ service descriptions are generated.

- 1a) **IMPLogWebService**(Wellbore wellbore_input, Date date_input, Drilling_Log dl_output, Lithology litho_output)
- 1b) **IMPLogWebService**(Wellbore wellbore_input, Date date_input, Drilling_Log dl_output)
- 2) **IMPLogWebService**(DrillHole dh_input, Date date_input, Drilling_Log dl_output, Mineral_Composition mc_output)
- 3) **IMPLogWebService**(DrillHole dh_input, Date date_input, Drilling_Log dl_output, Lithology litho_output)
- 4) **IMPLogWebService**(BoreHole bh_input, Date date_input, Drilling_Log dl_output, Mineral_Composition mc_output)
- 5) **IMPLogWebService**(BoreHole bh_input, Date date_input, Drilling_Log dl_output, Lithology mc_output)
- 6) **IMPLogWebService**(Hole hole_input, Date date_input, Drilling_Log dl_output, Mineral_Composition mc_output)
- 7) **IMPLogWebService**(Hole hole_input, Date date_input, Drilling_Log dl_output, Lithology litho_output)

In order to reduce the combinatorial complexity of this algorithm, we can explore the similarity degrees generating expanded service descriptions step by step.

4.2 Step-by-Step Generation of Service Descriptions

The equation (2) determines the number of similar service descriptions $\#q^D$ given a service description expansion in inputs $q_e^{D_e}$ and outputs $q_s^{D_s}$.

$$\#q^D = q_e^{D_e} * q_s^{D_s}, \text{ where } q_e^D = \begin{cases} \prod_{j=1}^{|I|} (a_{i_j}^{D_i} + 1) & \text{if } 1 \leq |I| \\ 0 & \text{otherwise} \end{cases} \text{ and } q_s^D = \begin{cases} \prod_{k=1}^{|O|} (d_{o_k}^{D_k} + 1) & \text{if } 1 \leq |O| \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Each $a_{i_j}^{D_i}$ represents a number of ancestors for each input i_j at the level D_i , where $i_j \in I$, and each $d_{o_k}^{D_k}$ is the number of descendents for each output o_k at the level D_k , where $o_k \in O$. D represents here the similarity degree, $D=e+s$, where e and s are the sums of the movements for inputs and outputs respectively (Table 2).

The similarity degrees defined in Section 3.2 can reduce significantly the number of expanded service descriptions only keeping those that have highest similarities. For a service specification with p parameters, all their combinations with a desired similarity degree should be found. For example, if we are looking for zero similarity degree, we obtain $(0_1, 0_2, 0_3, \dots, 0_p)$. To get the similarity equals to 1, there are many more options: $(1_1, 0_2, 0_3, \dots, 0_p)$, $(0_1, 1_2, 0_3, \dots, 0_p)$, ..., $(0_1, 0_2, 0_3, \dots, 1_p)$. The main difference of step-by-step expansion algorithm from the Algorithm 1 (Fig. 2) consists in obtaining only those parents and children which correspond to the levels defined for each parameter: if no service is found for the level, then the next one is analyzed.

For the level 1 from the example, the options are: $(-1_1, 0_2, 0_3, 0_4)$ and $(-0_1, 0_2, 0_3, 1_4)$, which will generate the descriptions corresponding to the descriptions 2) and 1a). The highest similarity level is calculated as $(\maxabs(p_1), \maxabs(p_2), \maxabs(p_3), \dots, \maxabs(p_p))$, which for the example equals to $(\maxabs(-3), \maxabs(0), \maxabs(0), \maxabs(1))=4$. The only one description corresponding to this level is 7).

5 DSC Based on Missing Information Matching

The algorithms described above help us to extend the descriptions of services for discovery. According to Kalasapur et al. [8], the DSC can be seen like an extension of discovery techniques. Let us consider a case when only missing information services are found. In this case, the ranking of missing information types (Table 1) is as follows: *Missing Outputs* > *Missing Inputs* > *Missing I&O*. The composition of a service sc_A , where $1 \leq i \leq |sc_A|$, includes four steps: service selection, generation of expanded service descriptions, service discovery and service matching evaluation.

1: Select services s_{ϕ_i} or sc_{ϕ_i} , which in regard to the client's request sc_i :

- Have greater number of outputs (missing outputs)
 - Have less number of inputs (missing inputs)
 - Have greater number of outputs and less number of inputs (missing I&O)
- Add s_{ϕ_i} or each element of sc_{ϕ_i} to sc_A .

2: Expand the client's query $s_{C_{i+1}}$ such that the inputs $I_{C_{i+1}}$ are the same as for the initial query, $I_{C_{i+1}} = I_C$ and the outputs $O_{C_{i+1}}$ are:

- $O_{C_{i+1}} = O_i$, missing outputs of the service s_{ϕ_i} or sc_{ϕ_i} in respect to sc_i ,
- $O_{C_{i+1}} = I_i$, missing inputs of the service s_{ϕ_i} or sc_{ϕ_i} in respect to sc_i .

This means that the new query will search for the service s_{ϕ_i} or sc_{ϕ_i} and $s_{\phi_{i+1}}$ or each element of $sc_{\phi_{i+1}}$ is added to sc_A .

3: Apply one of the algorithms of service discovery described above.

4: Matching evaluation: if there are services corresponding to sc_A and the matching between s_C and sc_A is:

- Missing outputs, missing inputs, and missing I&O, then return to step 1.
- Exact or leftover information, then the composition is completed.

The algorithm stops when one of the following conditions is achieved: there are no services offering missing I&O, composition attempts or time are expired.

6 Experimental Results

To test the feasibility of the proposed algorithms, firstly the number of similar descriptions to be generated was estimated. 70 WSs with 339 methods from available for use along a month in Xmethods [9] were chosen, which had parameter descriptions referring to specific domain ontology and thus permitted generating their

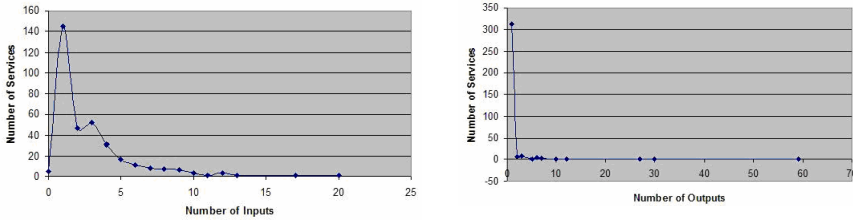


Fig. 3. Distributions of number of Inputs and Outputs per service

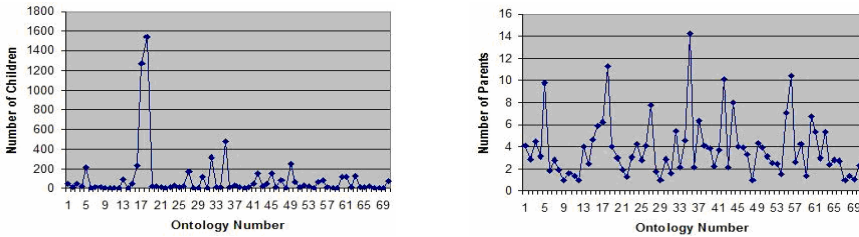


Fig. 4. Characteristics (descendents and antecedents) of the ontologies from [10]

semantic description. These services pertained to different categories like Web search, multimedia search, information coding, simulation, etc. The purpose of this experiment was to get an estimation of the number of inputs and outputs. Though the mode of inputs and outputs equals one (Fig. 3), the average numbers for Inputs and Outputs per service were calculated: $2.84 \cong 3$ and $1.58 \cong 2$ respectively.

Secondly, 70 OWL ontologies randomly selected from Swoogle [10] were analyzed to obtain the average number of ancestors and descendents per concept, in order to estimate the number of descriptions according to (1) and (2). As shown in Fig. 4, the averages for descendents and ancestors equal to $90.21 \cong 90$ and $3.83 \cong 4$ respectively. According to (1) the total number of similar service descriptions equals to 1'035,125.

However, for the second algorithm, calculated up to 10 degrees of similarity (d), the number of generated queries is the following: $d_0=0$, $d_1=20,720.25$, $d_2=51,726.25$, $d_3=103,512.5$, $d_4=207,025$, $d_5=269,132.25$, $d_6=207,025$, $d_7=103,512.5$, $d_8=51,726.25$, $d_9=20,720.25$, $d_{10}=0$ (Fig. 5) In other words, in the worst case (d5), the number of expanded descriptions is reduced to 25%.

Fig. 6 shows experimental results on time consumption of the proposed algorithm. The experiments have been carried out on a Notebook with processor Pentium M (1600MHz) and 512MB of RAM. Extrapolation of the obtained results show that for the step-by-step algorithm for the worse case (d5) the time of expanded service discovery will be about 30 minutes, time which can be considerably reduced if more powerful computer equipment is used or the query generation is realized in parallel. These results reflect the high impact of Protégé invocation on the total time.

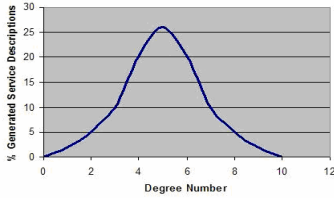


Fig. 5. A distribution of service descriptions generated based on the similarity degree

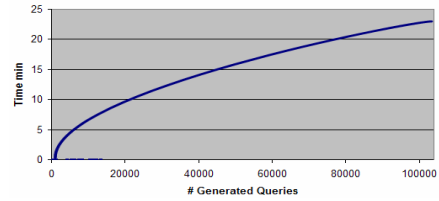


Fig. 6. Time consumption of the algorithm for service description expansion (one agent)

7 Conclusions

In the paper an approach to service discovery is described. It consists in novel types of service matching (exact, leftover and missing information) and two algorithms for extensive and step-by-step generation of similar OWL-S service descriptions. Once generated, these expanded descriptions can be sent to traditional syntactic UDDI, or can be filtered based on their semantic similarity in order to reduce inefficient calculations. The main advantages of the proposed approach are the following: i) it permits working with traditional syntactic UDDI, ii) permits generating from 25 to 40% more executable service descriptions than known methods [12]¹, and iii) at the same time, the step-by-step algorithm achieves the reduction of generated descriptions with low similarity degrees in the worst case of up to 75%. In the current work the proposed model, more particularly, matching with missing information and low similarity matching is applied for DSC.

References

1. Zhu, F., Mutka, M.W., Ni, L.M.: Service discovery in pervasive computing environments. In: *Pervasive Computing*, pp. 81–90. IEEE CS Press, Washington (2005)
2. Srinivasan, N., Paolucci, M., Sycara, K.: An Efficient Algorithm for OWL-S based Semantic Search in UDDI. In: Cardoso, J., Sheth, A.P. (eds.) *SWSWPC 2004*. LNCS, vol. 3387, pp. 96–110. Springer, Heidelberg (2005)
3. Dourdas, N., Zhu, X., Maiden, N.A.M., Jones, S., Zachos, K.: Discovering Remote Software Service that Satisfy Requirements: Patterns for Query Reformulation. In: Dubois, E., Pohl, K. (eds.) *CAiSE 2006*. LNCS, vol. 4001, pp. 239–254. Springer, Heidelberg (2006)
4. Ziembicki, J.I.: *Distributed Search in Semantic Web Service Discovery*. Master of Mathematics Thesis. University of Waterloo, Canada (2006)
5. Sánchez, C., Sheremetov, L.: A Model for Semantic Service Matching with Leftover and Missing Information. In: *8th Int. Conf. on Hybrid Intelligent Systems*, Barcelona, Spain, September 10–12, 2008. IEEE CS Press, Washington (2008)
6. Bhogal, J., Macfarlane, A., Smith, P.: A review of ontology based query expansion. *Information Processing & Management* 43(4), 866–886 (2007)

¹ Note, that though these methods generate more descriptions, most of them can't be executed since contain incomplete information.

7. Nardi, D., Brachman, R.: An Introduction to Description Logics. In: Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.) *The Description Logic Handbook: Theory, Implementation and Applications*, pp. 5–44. Cambridge Univ. Press, Cambridge (2003)
8. Kalasapur, S., Kumar, M., Shirazi, B.A.: Dynamic Service Composition in Pervasive Computing. In: *Parallel and Distributed Systems*, vol. 1818, pp. 907–918. IEEE CS Press, Washington (2007)
9. XMethods, <http://www.xmethods.net>
10. Swoogle the semantic web search engine and metadata service provider, <http://swoogle.umbc.edu>
11. Sycara, K., Paolucci, M., Ankolekar, A., Srinivasan, N.: Automated Discovery, Interaction and Composition of Semantic Web Services. *J. of Web Semantics* 1(1), 27–46 (2003)

Tuning Topical Queries through Context Vocabulary Enrichment: A Corpus-Based Approach*

Carlos M. Lorenzetti and Ana G. Maguitman

Grupo de Investigación en Recuperación de Información y Gestión del Conocimiento
LIDIA - Laboratorio de Investigación y Desarrollo en Inteligencia Artificial
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur, Av. Alem 1253, (B8000CPB) Bahía Blanca, Argentina
CONICET - Consejo Nacional de Investigaciones Científicas y Técnicas
phone: 54-291-4595135 fax: 54-291-4595136
{cml, agm}@cs.uns.edu.ar

Abstract. Context-based Web search has become an important research area and many strategies have been proposed to reflect contextual information in search queries. Despite the success of some of these proposals they still have serious limitations due to their inability to bridge the terminology gap existing between the user context description and the relevant documents' vocabulary. This paper presents a quantitative technique to learn vocabularies useful for describing the theme of a context under analysis. The enriched vocabulary allows the formulation of search queries to identify resources with higher precision than those identified using the initial vocabulary. Rigorous experimentation leads us to conclude that the proposed technique is superior to a baseline and other well-known query reformulation techniques.

1 Introduction

Context-based search is the process of seeking information related to a user's thematic context [5,11,8,15]. Meaningful automatic context-based search can only be achieved if the semantics of the terms in the context under analysis is reflected in the search queries. From a pragmatic perspective, terms acquire meaning from the way they are used and from their co-occurrence with other terms. Therefore, mining large corpora (such as the World Wide Web) guided by the user's context can help uncover the meaning of a user's information request.

An information request is usually initiated or generated within a task. For example, if the user is editing or reading a document on a specific topic, he may be willing to explore new material related to that topic. Topical queries can be formed using small sets of terms from the user's context. The implementation of a mechanism for the automatic generation of queries from a thematic context raises several questions: (1) Which terms will be more helpful to access relevant material? (2) How many terms should be used to form each query? (3) Is the current context vocabulary good enough to access the right information?

* This research work is supported by Agencia Nacional de Promoción Científica y Tecnológica (PICT 2005 Nro. 32373) and Universidad Nacional del Sur (PGI 24/ZN13).

Attempting to find the best subsets of terms to create appropriate queries is a combinatorial problem. The situation worsens when we deal with an open search space, i.e., when other terms that are not part of the current context vocabulary can be part of the queries. Willing to use terms that are not part of the current context is not an atypical situation when attempting to tune queries based on a small context description and a large external corpus. We can think of this query tuning process as a by-product of learning a better vocabulary to characterize the topic under analysis and the user's information needs.

The contribution of this work is a method for guiding the incremental exploration of new vocabularies with the purpose of tuning queries. The goal for the queries is to reflect contextual information and to effectively retrieve semantically related material when posed to a search interface.

2 Background

Query tuning is usually achieved by replacing or extending the terms of a query, or by adjusting the weights of a query vector. Relevance feedback is a query refinement mechanism used to tune queries based on the relevance assessments of the query's results. A driving hypothesis for relevance feedback methods is that it may be difficult to formulate a good query when the collection of documents is not known in advance, but it is easy to judge particular documents, and so it makes sense to engage in an iterative query refinement process. A typical relevance feedback scenario will involve the following steps:

Step 1: A query is formulated.

Step 2: The system returns an initial set of results.

Step 3: A relevance assessment on the returned results is issued (relevance feedback).

Step 4: The system computes a better representation of the information needs based on this feedback.

Step 5: The system returns a revised set of results.

Depending on the level of automation of step 3 we can distinguish three forms of feedback:

- **Supervised Feedback:** requires explicit feedback, which is typically obtained from users who indicate the relevance of each of the retrieved documents.
- **Unsupervised Feedback:** it applies blind relevance feedback, and typically assumes that the top k documents returned by a search process are relevant.
- **Semi-supervised Feedback:** the relevance of a document is inferred by the system. A common approach is to monitor the user behavior (e.g., documents selected for viewing or time spent viewing a document). Provided that the information seeking process is performed within a thematic context, another automatic way to infer the relevance of a document is by computing the similarity of the document to the user's current context.

The best-known algorithm for relevance feedback has been proposed by Rocchio [17]. Given an initial query vector \vec{q} a modified query \vec{q}_m is computed as follows:

$$\vec{q}_m = \alpha \vec{q} + \beta \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \sum_{\vec{d}_j \in D_n} \vec{d}_j.$$

where D_r and D_n are the sets of relevant and non-relevant documents respectively and α , β and γ are tuning parameters. A common strategy is to set α and β to a value greater than 0 and γ to 0, which yields a positive feedback strategy. When user relevance judgments are unavailable, the set D_r is initialized with the top k retrieved documents and D_n is set to \emptyset . This yields an unsupervised relevance feedback method.

Several successors of the Rocchio's method have been proposed with varying success. One of them is selective query expansion [2], which monitors the evolution of the retrieved material and is disabled if query expansion appears to have a negative impact on the retrieval performance. Other successors of the Rocchio's method use an external collection different from the target collection to identify good terms for query expansion. The refined query is then used to retrieve the final set of documents from the target collection [9]. A successful generalization of the Rocchio's method is the Divergence from Randomness mechanism with Bose-Einstein statistics (Bo1-DFR) [1]. To apply this model, we first need to assign weights to terms based on their informativeness. This is estimated by the divergence of its distribution in the top-ranked documents from a random distribution as follows:

$$w(t) = t f_x \cdot \log_2 \frac{1 + P_n}{P_n} + \log_2(1 + P_n)$$

where $t f_x$ is the frequency of the query term in the top-ranked documents and P_n is the proportion of documents in the collection that contain t . Finally, the query is expanded by merging the most informative terms with the original query terms.

The main problem of the above query tuning methods is that their effectiveness is correlated with the quality of the top ranked documents returned by the first-pass retrieval. On the other hand, if a thematic context is available, the query refinement process can be guided by computing an estimation of the quality of the retrieved documents. This estimation can be used to predict which terms can help refine subsequent queries.

During the last years several techniques that formulate queries from the user context have been proposed [5,8]. Limited work, however, has been done on semi-supervised methods that simultaneously take advantage of the user context and results returned from a corpus to refine queries. Next section presents our proposal to tune topical queries based on the analysis of the terms found in the user context and in the incrementally retrieved results.

3 A Semi-supervised Method for Query Reformulation

Much work has addressed the problem of computing the informativeness of a term across a corpus (e.g., [11,16]) and a good deal of research has focused on computing the descriptive and discriminating power of a term in a document with respect to a corpus

(e.g., [18]). All this work, however, has been done based on a predefined collection of documents and independently from a thematic context. In [12] we proposed to study the descriptive and discriminating power of a term based on its distribution across the topics of pages returned by a search engine. In that proposal the search space is the full Web and the analysis of the descriptive or discriminating power of a term is limited to a small collection of documents—incremental retrievals—that is built up over time and changes dynamically. Unlike traditional information retrieval schemes, which analyze a predefined collection of documents and search that collection, our methods use limited information to assess the importance of terms and documents as well as to manage decisions about which terms to retain for further analysis, which ones to discard, and which additional queries to generate.

To distinguish between topic descriptors and discriminators we argue that *good topic descriptors* can be found by looking for terms that occur often in documents related to the given topic. On the other hand, *good topic discriminators* can be found by looking for terms that occur only in documents related to the given topic. Both topic descriptors and discriminators are important as query terms. Because topic descriptors occur often in relevant pages, using them as query terms may improve recall. Similarly, good topic discriminators occur primarily in relevant pages, and therefore using them as query terms may improve precision.

3.1 Computing Descriptive and Discriminating Power

As a first approximation to compute descriptive and discriminating power, we begin with a collection of m documents and n terms. As a starting point we build an $m \times n$ matrix \mathbf{H} , such that $\mathbf{H}[i, j] = k$ if k is the number of occurrences of term t_j in document d_i . In particular we can assume that one of the documents (e.g., d_0) corresponds to the initial user context.

The matrix \mathbf{H} allows us to formalize the notions of good descriptors and good discriminators. We define *descriptive power of a term in a document* as a function $\lambda : \{d_0, \dots, d_{m-1}\} \times \{t_0, \dots, t_{n-1}\} \rightarrow [0, 1]$:

$$\lambda(d_i, t_j) = \frac{\mathbf{H}[i, j]}{\sqrt{\sum_{k=0}^{n-1} (\mathbf{H}[i, k])^2}}$$

Note that λ can be regarded as a version of matrix \mathbf{H} normalized by row (i.e, by document).

If we adopt $s(k) = 1$ whenever $k > 0$ and $s(k) = 0$ otherwise, we can define the *discriminating power of a term in a document* as a function $\delta : \{t_0, \dots, t_{n-1}\} \times \{d_0, \dots, d_{m-1}\} \rightarrow [0, 1]$:

$$\delta(t_i, d_j) = \frac{s(\mathbf{H}[j, i])}{\sqrt{\sum_{k=0}^{m-1} s(\mathbf{H}[k, i])}}$$

In this case δ can be regarded as a transposed version of matrix \mathbf{H} normalized by column (i.e, by term).

Our current goal is to learn a better characterization of the user needs. Therefore rather than extracting descriptors and discriminators directly from the user context, we

want to extract them from *the topic* of the user context. This requires an incremental method to characterize the topic of the user context, which is done by identifying documents that are similar to the user current context. Assume the user context and the retrieved documents are represented as document vectors in term space. To determine how similar two documents d_i and d_j are, we adopt the IR cosine similarity [3]. This measure is defined as a function $\sigma : \{d_0, \dots, d_{m-1}\} \times \{d_0, \dots, d_{m-1}\} \rightarrow [0, 1]$:

$$\sigma(d_i, d_j) = \sum_{k=0}^{n-1} [\lambda(d_i, t_k) \cdot \lambda(d_j, t_k)].$$

We formally define the *term descriptive power in the topic of a document* as a function $\Lambda : \{d_0, \dots, d_{m-1}\} \times \{t_0, \dots, t_{n-1}\} \rightarrow [0, 1]$. We set $\Lambda(d_i, t_j) = 0$ if $\sum_{\substack{k=0 \\ k \neq i}}^{m-1} \sigma(d_i, d_k) = 0$. Otherwise we define $\Lambda(d_i, t_j)$ as follows:

$$\Lambda(d_i, t_j) = \frac{\sum_{\substack{k=0 \\ k \neq i}}^{m-1} [\sigma(d_i, d_k) \cdot [\lambda(d_k, t_j)]^2]}{\sum_{\substack{k=0 \\ k \neq i}}^{m-1} \sigma(d_i, d_k)}.$$

Thus, the descriptive power of a term t_j in the topic of a document d_i is a measure of the quality of t_j as a descriptor of documents similar to d_i .

Analogously, we define the *discriminating power of a term in the topic of a document* as a function $\Delta : \{t_0, \dots, t_{n-1}\} \times \{d_0, \dots, d_{m-1}\} \rightarrow [0, 1]$ calculated as follows:

$$\Delta(t_i, d_j) = \sum_{\substack{k=0 \\ k \neq j}}^{m-1} [[\delta(t_i, d_k)]^2 \cdot \sigma(d_k, d_j)].$$

Thus the discriminating power of term t_i in the topic of document d_j is an average of the similarity of d_j to other documents discriminated by t_i . For a worked example showing the results of computing topic descriptors and discriminators see [10].

3.2 An Incremental Mechanism to Tune Topical Queries

Our proposal is to approximate the terms' descriptive and discriminating power for the thematic context under analysis with the purpose of generating good queries.

Our approach adapts the typical relevance feedback mechanism to account for a thematic context \mathcal{C} as follows:

Step 1: A query is formulated based on \mathcal{C} .

Step 2: The system returns an initial set of results.

Step 3: Repeat for at least v iterations or until no improvements are registered

Step 3.1: A relevance assessment on the returned results is issued based on \mathcal{C} .

Step 3.2: After a certain number of trials and depending on the relevance assessments, the system computes a better representation of the thematic context (phase change).

Step 3.3: The system formulates new queries and returns a revised set of results.

In order to learn better characterizations of the thematic context, the system undergoes a series of phases. At the end of each phase, the context characterization is updated with new learned material. Each phase evolves through a sequence of trials, where each trial consists in the formulation of a set of queries, the analysis of the retrieved results, the adjustment of the terms' weights, and the discovery of new potentially useful terms. For a given phase \mathcal{P}_i , the context is represented by a set of weighted terms. Let $w^{\mathcal{P}_i}(t, \mathcal{C})$ be an estimation of the importance of term t in context \mathcal{C} during phase i . If t occurs in the initial context, then the value $w^{\mathcal{P}_0}(t, \mathcal{C})$ is initialized as the normalized frequency of term t in \mathcal{C} , while the weight of those terms that are not part of \mathcal{C} are assumed to be 0.

Let $w_{\Lambda}^{(i,j)}(t, \mathcal{C})$ and $w_{\Delta}^{(i,j)}(t, \mathcal{C})$ be an estimation of the descriptive and discriminating power of term t for context \mathcal{C} at trial j of phase i . These values are incrementally computed as follows:

$$w_{\Lambda}^{(i,j+1)}(t, \mathcal{C}) = \alpha.w_{\Lambda}^{(i,j)}(t, \mathcal{C}) + \beta.\Lambda^{(i,j)}(t, \mathcal{C}).$$

$$w_{\Delta}^{(i,j+1)}(t, \mathcal{C}) = \alpha.w_{\Delta}^{(i,j)}(t, \mathcal{C}) + \beta.\Delta^{(i,j)}(t, \mathcal{C}).$$

We assume $w_{\Lambda}^{(i,0)}(t, \mathcal{C}) = w_{\Delta}^{(i,0)}(t, \mathcal{C}) = 0$ and use the results returned during each trial j to compute $\Lambda^{(i,j)}(t, \mathcal{C})$ and $\Delta^{(i,j)}(t, \mathcal{C})$, the descriptive and discriminating power of term t for the topic of \mathcal{C} . To form queries during phase i we implemented a roulette selection mechanisms where the probability of choosing a particular term t to form a query is proportional to $w^{\mathcal{P}_i}(t, \mathcal{C})$. Roulette selection is a technique typically used by Genetic Algorithms [7] to choose potentially useful solutions for recombination, where the fitness level is used to associate a probability of selection. This approach resulted in a non-deterministic exploration of term space that favored the fittest terms.

The system monitors the effectiveness achieved at each iteration. In our approach we use *novelty-driven similarity* introduced in section 4 as an estimation of the retrieval effectiveness. If after a window of u trials the retrieval effectiveness has not crossed a given threshold μ (i.e., no significant improvements are observed after certain number of trials), the system forces a phase change to explore new potentially useful regions of the vocabulary landscape. A phase change can be regarded as a vocabulary leap, which can be thought of as a significant transformation (typically an improvement) of the context characterization. If a phase change takes effect during trial j , the value of $w_{\Lambda}^{\mathcal{P}_i}(t, \mathcal{C})$ is set to $w_{\Lambda}^{(i,j)}(t, \mathcal{C})$ and $w_{\Delta}^{\mathcal{P}_i}(t, \mathcal{C})$ is set to $w_{\Delta}^{(i,j)}(t, \mathcal{C})$. To reflect the phase change in the new characterization of the thematic context, the weight of each term t is updated as follows:

$$w^{\mathcal{P}_{i+1}}(t, \mathcal{C}) = \gamma.w^{\mathcal{P}_i}(t, \mathcal{C}) + \zeta.w_{\Lambda}^{\mathcal{P}_i}(t, \mathcal{C}) + \xi.w_{\Delta}^{\mathcal{P}_i}(t, \mathcal{C}).$$

These weights are then used to generate new queries during the sequence of trials at phase $i + 1$.

4 Evaluation

The goal of this section is to compare the proposed method against two other methods. The first is a baseline that submits queries directly from the thematic context and does

not apply any refinement mechanism. The second method used for comparison is the Bo1-DFR described in section 2.

To perform our tests we used 448 topics from the Open Directory Project (ODP). The topics were selected from the third level of the ODP hierarchy. A number of constraints were imposed on this selection with the purpose of ensuring the quality of our test set. The minimum size for each selected topic was 100 URLs and the language was restricted to English. For each topic we collected all of its URLs as well as those in its subtopics. The total number of collected pages was more than 350K. The Terrier framework [14] was used to index these pages and to run our experiments.

In our tests we used the ODP description of each selected topic to create an initial context description \mathcal{C} . The proposed algorithm was run for each topic for at least $v = 100$ iterations, with 10 queries per iteration and retrieving 10 results per queries.

The descriptor and discriminator lists at each iteration were limited to up to 100 terms each. The other parameters in our algorithm were set as follows: $u = 10$, $\alpha=0.5$, $\beta=0.5$, $\gamma=0.33$, $\zeta=0.33$, $\xi=0.33$, $\mu=0.2$. In addition, we used the stopword list provided by Terrier, Porter stemming was performed on all terms and none of the query expansion methods offered by Terrier was applied.

In order to compare the implemented methods we used three measures of query performance:

- Novelty-driven similarity: this measure of similarity is based on σ but disregards the terms that form the query, overcoming the bias introduced by those terms and favoring the exploration of new material. Given a query q and documents d_i and d_j , the novelty-driven similarity measure is defined as $\sigma^N(\mathbf{q}, d_i, d_j) = \sigma(d_i - \mathbf{q}, d_j - \mathbf{q})$. The notation $d_i - \mathbf{q}$ stands for the representation of the document d_i with all the values corresponding to the terms from query \mathbf{q} set to zero. The same applies to $d_j - \mathbf{q}$.
- Precision: this performance evaluation measures the fraction of retrieved documents which are known to be relevant, i.e., $\text{Precision} = |A \cap R|/|A|$, where R and A are the relevant and answer set respectively. The relevant set for each analyzed topic was set as the collection of its URLs as well as those in its subtopics.
- Semantic Precision: other topics in the ontology could be semantically similar (and therefore partially relevant) to the topic of the given context. Therefore, we propose a measure of semantic precision defined as $\text{Precision}^S = \sum_{p \in A} \sigma^S(t(\mathcal{C}), t(p))/|A|$, where $t(\mathcal{C})$ is the ODP topic associated with the description used as the initial context, $t(p)$ is the topic of page p and $\sigma^S(t(\mathcal{C}), t(p))$ is the semantic similarity between these two topics. To compute σ^S we used a semantic similarity measure for generalized ontologies proposed by Maguitman et al. [13].

The charts in figure 1 compare the performance of queries for each tested method using novelty-driven similarity and precision. Each of the 448 topics corresponds to a trial and is represented by a point. The point's vertical coordinate (z) corresponds to the performance of the incremental method, while the point's other two coordinates (x and y) correspond to the baseline and the Bo1-DFR methods. In addition we can observe the projection of each point on the x-y, x-z and y-z planes. For the x-z plane, the points

¹ <http://dmoz.org>

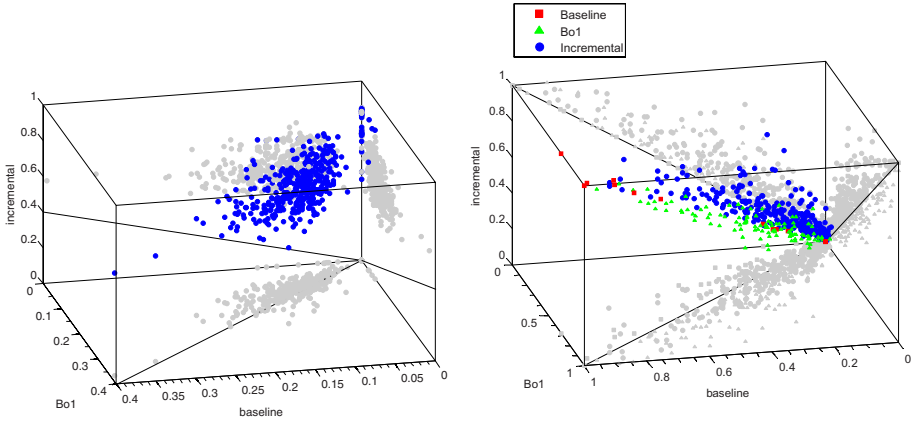


Fig. 1. A comparison of the three tested methods based on novelty-driven similarity (left) and precision (right)

above the diagonal correspond to cases in which the incremental method is superior to the baseline. Similarly, for the y-z plane, the points above the diagonal correspond to cases in which the incremental method is superior to Bo1-DFR. The x-y plane compares the performance of the baseline against Bo1-DFR.

It is interesting to note that for all the tested cases the incremental method was superior to the baseline and the Bo1-DFM method in terms of novelty-driven similarity. This highlights the usefulness of evolving the context vocabularies to discover good query terms. For the precision metric, the incremental method was strictly superior to the other two methods for 66.96% of the evaluated topics. Bo1-DFR was the best method for 24.33% of the topics and the baseline performed as well as one of the other two other methods for 8.70% of the topics. For the semantic precision metric (not shown for space limitations) the incremental method was strictly superior to the other methods for 65.18% of the topics, Bo1-DFR was superior for 27.90% of the topics and the baseline performed as well as one of the other two methods for 6.92% of the topics.

The next table presents the means and confidence intervals of the methods’ performance based on σ^N , Precision and Precision^S. This comparison table shows that the improvements achieved by the incremental method with respect to the other methods are statistically significant.

	N	σ^N		Precision		Precision ^S	
		mean	95% C.I.	mean	95% C.I.	mean	95% C.I.
Baseline	448	0.087	[0.0822;0.0924]	0.266	[0.2461;0.2863]	0.553	[0.5383;0.5679]
Bo1-DFR	448	0.075	[0.0710;0.0803]	0.307	[0.2859;0.3298]	0.590	[0.5750;0.6066]
Incremental	448	0.597	[0.5866;0.6073]	0.354	[0.3325;0.3764]	0.622	[0.6068;0.6372]

5 Conclusions

The vocabulary problem is a main challenge in human-system communication. In this paper we propose a solution to the semantic sensitivity problem, that is the limitation that arises when documents with similar context but different term vocabulary won't be associated, resulting in a false negative match. Our method operates by incrementally learning better vocabularies from a large external corpus such as the Web.

Other corpus-based approaches have been proposed to address the semantic sensitivity problem. For example, latent semantic analysis [6] applies singular value decomposition to reduce the dimensions of the term-document space, harvesting the latent relations existing between documents and between terms in large text corpora. Another corpus-based technique that has been applied to estimate semantic similarity is PMI-IR [20]. This information retrieval method is based on pointwise mutual information, which measures the strength of association between two elements (e.g., terms) by contrasting their observed frequency against their expected frequency. Differently from our approaches, these techniques are not based on an incrementally refined query submission process. Instead, they use a predefined collection of document to identify latent semantic relations. In addition, these techniques do not distinguish between the notions of topic descriptors and topic discriminators. The techniques for query term selection proposed in this paper share insights and motivations with other methods for query expansion and refinement [19,4]. However, systems applying these methods differ from our framework in that they support this process through query or browsing interfaces requiring explicit user intervention, rather than formulating queries automatically.

In this paper we have shown that by implementing an incremental context refinement method we can perform better than a baseline method, which submit queries directly from the initial context, and to the Bo1-DFR method, which does not refine queries based on context. This points to the usefulness of simultaneously taking advantage of the terms in the current thematic context and an external corpus to learn better vocabularies and to automatically tune queries.

References

1. Amanti, G.: Probabilistics Models for Information Retrieval based on Divergence from Randomness. PhD thesis, Department of Computing Science, University of Glasgow, UK (2003)
2. Amati, G., Carpineto, C., Romano, G.: Query difficulty, robustness and selective application of query expansion. In: *Advances in Information Retrieval, 26th European Conference on IR research*, pp. 127–137. Springer, Heidelberg (2004)
3. Baeza-Yates, R., Ribeiro-Neto, B.: *Modern Information Retrieval*. Addison-Wesley, Reading (1999)
4. Billerbeck, B., Scholer, F., Williams, H.E., Zobel, J.: Query expansion using associated queries. In: *Proceedings of the twelfth international conference on Information and knowledge management*, pp. 2–9. ACM Press, New York (2003)
5. Budzik, J., Hammond, K.J., Birnbaum, L.: Information access in context. *Knowledge based systems* 14(1-2), 37–53 (2001)
6. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. *Journal of the American Society of Information Science* 41(6), 391–407 (1990)

7. Holland, J.H.: *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor (1975)
8. Kraft, R., Chang, C.C., Maghoul, F., Kumar, R.: Searching with context. In: WWW 2006: Proceedings of the 15th international conference on World Wide Web, pp. 477–486. ACM, New York (2006)
9. Kwok, K.L., Chan, M.: Improving two-stage ad-hoc retrieval for short queries. In: SIGIR 1998: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 250–256. ACM, New York (1998)
10. Lorenzetti, C.M., Cecchini, R.L., Maguitman, A.G.: Intelligent methods for information access in context: The role of topic descriptors and discriminators. In: VIII Workshop de Agentes y Sistemas Inteligentes - CACIC 2007: XIII Congreso Argentino de Ciencias de la Computación, Corrientes, Argentina (October 2007)
11. Maguitman, A., Leake, D., Reichherzer, T.: Suggesting novel but related topics: towards context-based support for knowledge model extension. In: IUI 2005: Proceedings of the 10th international conference on Intelligent user interfaces, pp. 207–214. ACM Press, New York (2005)
12. Maguitman, A., Leake, D., Reichherzer, T., Menczer, F.: Dynamic extraction of topic descriptors and discriminators: Towards automatic context-based topic search. In: Proceedings of the Thirteenth Conference on Information and Knowledge Management (CIKM). ACM Press, Washington (2004)
13. Maguitman, A.G., Menczer, F., Roinestad, H., Vespignani, A.: Algorithmic detection of semantic similarity. In: WWW 2005: Proceedings of the 14th international conference on World Wide Web, pp. 107–116. ACM, New York (2005)
14. Ounis, I., Lioma, C., Macdonald, C., Plachouras, V.: Research directions in Terrier: a search engine for advanced retrieval on the web. In: Baeza-Yates, R., et al. (eds.) *Novatica/UPGRADE Special Issue on Web Information Access*, February 2007, vol. VIII(1), pp. 49–56 (2007)
15. Ramirez, E.H., Brena, R.F.: Semantic contexts in the internet. In: LA-WEB 2006: Proceedings of the Fourth Latin American Web Congress, Washington, DC, USA, pp. 74–81. IEEE Computer Society, Los Alamitos (2006)
16. Rennie, J.D.M., Jaakkola, T.: Using term informativeness for named entity detection. In: SIGIR 2005: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 353–360. ACM, New York (2005)
17. Rocchio, J.J.: Relevance feedback in information retrieval. In: Salton, G. (ed.) *The Smart retrieval system - experiments in automatic document processing*, pp. 313–323. Prentice-Hall, Englewood Cliffs (1971)
18. Salton, G., Buckley, C.: Term weighting approaches in automatic text retrieval. *Information Processing and Management* 24(5), 513–523 (1988)
19. Scholer, F., Williams, H.E.: Query association for effective retrieval. In: Proceedings of the eleventh international conference on Information and knowledge management, pp. 324–331. ACM Press, New York (2002)
20. Turney, P.D.: Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In: Flach, P.A., De Raedt, L. (eds.) *ECML 2001. LNCS (LNAI)*, vol. 2167, pp. 491–502. Springer, Heidelberg (2001)

Measuring Heterogeneity between Web-Based Agent Communication Protocols

Maricela Bravo¹ and José Velazquez²

¹Morelos State Polytechnic University, Cuauhnhuac 566, Texcal,
Morelos, México, CP 62550
mbravo@upemor.edu.mx

²Electrical Research Institute, Reforma 113, Palmira,
Morelos, México, CP 62490
jconrado@iie.org.mx

Abstract. Communication between multiple agents is essential to achieve goals, cooperation, and negotiation to take decisions for mutual benefit. Nowadays there is a growing interest in automating communication processes between different agents in dynamic Web-based environments. However, when agents are deployed and integrated in open and dynamic environments, detailed syntax and semantics of their particular language implementations differ, causing the problem of communication heterogeneity. Therefore, it is necessary to measure heterogeneity among all participating agents and the number of required translations when heterogeneous agents are involved in communications. In this paper we present a set of measures with the objective to evaluate the minimal computational requirements before implementing a translation approach. Our measures are based on set theory, which has proved to be a good representation formalism in other areas. We showed how to use the set of measures for two highly heterogeneous set of agents.

Keywords: Agent communication protocols, measuring heterogeneity.

1 Introduction

Communication between multiple agents is essential to achieve goals, cooperation, and negotiation to take decisions for mutual benefit. Nowadays there is a growing interest in automating communication processes between different agents in dynamic Web-based environments. Traditionally autonomous agents have been implemented independently of each other in local environments, where agents exchange messages following a protocol based on standard language specifications. However, when agents are deployed and integrated in open and dynamic environments, detailed syntax and semantics of their particular language implementations differ, causing the problem of *communication heterogeneity*.

There are two traditional solutions to support communication interoperability: one is the use of a common communication protocol and the second is a translation approach with the use of a shared ontology. The use of a common communication protocol consists of defining syntax and formal semantics of primitives for all participating agents,

the main drawback of this approach is the lack of flexibility, and the required time and effort for coding each agent every time that it is going to be deployed on a different agent community, with different communication rules; therefore this solution is not suitable for dynamic environments.

The second approach consists of automatic translation of exchanged messages by means of a shared ontology populated with all the communication primitives. However, there are certain computational considerations when selecting this approach, because it requires the use of a shared ontology and a translator. To help the developer to decide which approach better suits its requirements, it is necessary to measure heterogeneity among all participating agents and the number of required translations when many heterogeneous agents are involved in translation. In this paper we present a set of measures that will provide such information.

The rest of the paper is organized as follows. In section two we present related work with the subject of this research. In section three we describe a set of measures to evaluate the main requirements of a translation approach. In section four we apply the measures to a case study, in order to clarify the measures usage. In section five we conclude and give future direction of this work.

2 Related Works

Many authors agree on the importance of communication interoperability and the correct interpretation of exchanged messages in multi-agent systems. Malucelli and Oliveira [1] stated that a critical factor for the efficiency of communication processes and the success of potential settlements is an agreement between parties about how the issues of a communication are represented and what this representation means to each of the parties. In [2] authors explain that interoperability is about effective use of system services. They argue that the most important precondition to achieve interoperability is to ensure that the message sender and receiver share the same understanding of the data in the message and the same expectation of the effect of the message. Rueda [3] argues that the success of an agent application depends on the communication language, allowing agents to interact and share knowledge. In [4] authors state that communication deals with how to represent agent's requirements and constraints on a product and service and how to convey intentions by passing messages between parties.

There has been a common interest among multiple researchers, in providing communication interoperability between agents using an ontology to support translation. Gruber [5] presented Ontolingua, a system for describing ontologies in a compatible form with multiple representation languages. He used Ontolingua to translate definitions written in KIF (Knowledge Interchange Format) into different representation languages. Willmott et al. [6] presented an abstract ontology representation (AOR) to capture models of communication to deal with the interoperability problem between multiple agents. Hübner [4] described Saci (Simple Agent Communication Infrastructure) a tool for programming communication between multiple agents using KQML messages. Furlan et al. [7] designed a CORBA interface to Saci to support communication interoperability between Saci-based agents and CORBA-based agents. In [8] the author proposed a translation approach to facilitate agent communication, where agents use partially shared ontologies for multi-agent communication. In [9], authors

propose a common ontology for defining semantics for agent communication languages, based on public mental attitudes. In [10], we developed and presented a shared ontology to support communication in electronic systems, invoking a translator only when necessary.

In the revised works we can appreciate that many solutions have been proposed to support communication interoperability. However, before the selection of one of these solutions, we need to obtain more information relative to the specific scenario, which is going to be implemented, because system integrators may choose a novel solution approach, but if we do not evaluate the computational requirements we may be choosing a heavy and complex solution.

3 Measuring Heterogeneity

Communication among multiple agents in Web-based environments is executed through the exchange of messages following a protocol. A protocol is a sequence of exchanged messages conforming to a set of shared rules. For this work we are considering that a message has the following elements:

<Sender, Receiver, Primitive, Parameters, Other data>

Where

- *Sender* identifies the agent that is issuing the message,
- *Receiver* is the target agent, which will receive and analyze the incoming message,
- *Primitive* is a basic communication act, which has syntax, semantics and pragmatics.
- *Parameters* are the rest of input data, these parameters depend on the primitive.
- *Other data* is left for additional information.

To measure heterogeneity between multiple agents we considered the set of communication primitives of each agent, and designed a set of measures, which will be described next.

a) Universal Set of Communication Primitives

Given a set of heterogeneous agents

$$A = \{a_1, a_2, a_3, \dots, a_n\},$$

the universal set of communication primitives is represented by

$$CP = \{CPa_1, CPa_2, \dots, CPa_n\} \quad (1)$$

where $CPa_n = \{p_1, p_2, p_3, \dots, p_i\}$, is the set of communication primitives of agent n .

b) Number of Communication links

Considering a set of n agents, the possible number of peer to peer communication links among them is n^2 . However, as we are evaluating heterogeneity, we need to extract the number of communication links where agents are equal, which is n . We also considered that a communication link between agents (a, b) has the same heterogeneity as a communication link of agents (b, a) , thus we reduced the number of different communication links dividing by 2.

The number of different communication links between n agents is given by

$$CL = (n^2 - n) / 2 \quad (2)$$

c) *Set of Different Communication links*

Considering a set of agents, the set of different communication links is given by

$$DCL = \{ (a_1, a_2), (a_1, a_3), \dots, (a_i, a_j) \}, \quad (3)$$

d) *Number of communication primitives*

The total number of communication primitives is obtained from the union operation of all sets of communication primitives.

$$CPT = |CPa_1 \cup CPa_2 \cup \dots \cup CPa_n| \quad (4)$$

e) *Number of different communication primitives*

The total number of different communication primitives results from extracting the intersection of common communication primitives from CPT .

$$DCPT = |CPa_1 \cup CPa_2 \cup \dots \cup CPa_n| - |CPa_1 \cap CPa_2 \cap \dots \cap CPa_n| \quad (5)$$

f) *Level of heterogeneity*

The level of heterogeneity results from dividing $DCPT$ by CPT , which is the ratio that will serve as an indicator for evaluating heterogeneity.

$$\text{Level of heterogeneity} = DCPT / CPT \quad (6)$$

g) *Number of required translations*

Considering that for each pair (x_i, y_i) of communication links the necessary translations among them is the number of communication primitives that are unknown for each agent. This is, the number of translations that agent x_i needs to be translated is the set of communication primitives from agent y_i , minus the set of communication primitives that are common for both. For this formula we considered the sum of required translations of each agent independently, because translation is required independently of the translations of other agents.

$$\sum \forall (x_i, y_i) \in DCL = |x_i \cup y_i| - |x_i \cap y_i| \quad (7)$$

4 Case Studies

In this section, we present two case studies to apply the set of measures and evaluate the resulting values. We selected the set of communication primitives from the agents presented in [11]. We utilized three agents for the first case, and eight agents for the second, this decision was taken to discriminate if the number of agents has a considerable impact on the heterogeneity among them.

Case one. Measuring Communication Heterogeneity among Three Agents

Given a set $A = \{ a_1, a_2, a_3 \}$, of three autonomous agents developed independently of each other, deployed on the Web, with their respective set of communication primitives as follows:

$CPa_1 = \{Initial_Offer, RFQ, Accept, Reject, Offer, Counter_Offer\}$

$CPa_2 = \{CFP, Propose, Accept, Terminate, Reject, Acknowledge, Modify, Withdraw\}$

$CPa_3 = \{Requests_Add, Authorize_Add, Require, Demand, Accept, Reject, Unable, Require-for, Insist_for, Demand_for\}$

We first calculate the number of different communication links between them with $n = 3$, using Formula 2.

$$CL = (3^2-3)/2 = 3$$

The resulting set of different communication links is:

$$DCL = \{ (a_1, a_2), (a_1, a_3), (a_2, a_3) \}$$

Using Formula 4, we calculate the total number of communication primitives.

$CPa_1 \cup CPa_2 \cup CPa_3 = \{ Initial_Offer, RFQ, Accept, Reject, Offer, Counter_Offer, CFP, Propose, Terminate, Acknowledge, Modify, Withdraw, Requests_Add, Authorize_Add, Require, Demand, Unable, Require-for, Insist_for, Demand_for \}$

$$CPT = |CPa_1 \cup CPa_2 \cup CPa_3|$$

$$CPT = 20$$

Using Formula 5 we calculate the total number of different communication primitives.

$$|CPa_1 \cup CPa_2 \cup CPa_3| = 20$$

$$|CPa_1 \cap CPa_2 \cap CPa_3| = 2$$

$$DCPT = 18$$

We apply Formula 6 to calculate the level of heterogeneity for this set of heterogeneous agents.

$$Level\ of\ heterogeneity = 18 / 20 = 0.9$$

The level of heterogeneity results too high for these agents. We should analyze if a translation approach will have higher computational requirements to support communication interoperability. To select a translation approach we need to measure the number of individuals in the shared ontology. For our case study the number of individuals is equal to the total number of communication primitives, which is $CPa_1 \cup CPa_2 \cup CPa_3 = 20$.

There are two common primitives for all agents: *accept* and *reject*. To calculate the number of required translations, we considered the set of different communication

links $\mathbf{DCL} = \{ (a_1, a_2), (a_1, a_3), (a_2, a_3) \}$, and the worst case where each agent needs the translation of all communication primitives from the other agents in a peer to peer communication. Using Formula 7, the number of required translations for agents a_1, a_2, a_3 is given by:

$$\begin{aligned} \sum \forall (x_i, y_i) \in \mathbf{DCL} &= |\mathbf{CP}x_i \cup \mathbf{CP}y_i| - |\mathbf{CP}x_i \cap \mathbf{CP}y_i| \\ |\mathbf{CP}a_1 \cup \mathbf{CP}a_2| - |\mathbf{CP}a_1 \cap \mathbf{CP}a_2| &= 12 - 2 = 10 \\ |\mathbf{CP}a_1 \cup \mathbf{CP}a_3| - |\mathbf{CP}a_1 \cap \mathbf{CP}a_3| &= 14 - 2 = 12 \\ |\mathbf{CP}a_2 \cup \mathbf{CP}a_3| - |\mathbf{CP}a_2 \cap \mathbf{CP}a_3| &= 16 - 2 = 14 \\ \sum \forall (x_i, y_i) \in \mathbf{DCL} &= \mathbf{36} \end{aligned}$$

The minimal number of required translations in the worst case, for this scenario is 36, with a 0.9 level of heterogeneity. For minimal we mean that the translator translates only once for a communication session, considering that each participating agent has the ability to temporally learn a communication primitive that has been translated. The worst case represents the communication session into which all the different communication primitives will be issued, then causing the translator to execute the maximal number of translations for this scenario. We consider this scenario as a high heterogeneity case. However, the implementation of the translation approach will not be too costly, because the number of required instances in the ontology is 20, and the number of required translations is 36.

Case two. Measuring Communication Heterogeneity among eight Agents

Given a set $B = \{ a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8 \}$, of eight autonomous agents developed independently of each other and deployed on the Web, and their respective set of communication primitives, as follows.

$\mathbf{CP}a_1 = \{ \text{Initial_Offer}, \text{RFQ}, \text{Accept}, \text{Reject}, \text{Offer}, \text{Counter_Offer} \}$

$\mathbf{CP}a_2 = \{ \text{CFP}, \text{Propose}, \text{Accept}, \text{Terminate}, \text{Reject}, \text{Acknowledge}, \text{Modify}, \text{Withdraw} \}$

$\mathbf{CP}a_3 = \{ \text{Requests_Add}, \text{Authorize_Add}, \text{Require}, \text{Demand}, \text{Accept}, \text{Reject}, \text{Unable}, \text{Require-for}, \text{Insist-for}, \text{Demand-for} \}$

$\mathbf{CP}a_4 = \{ \text{accept-offer}, \text{what-is-price}, \text{what-is-item}, \text{add-sell-agent}, \text{add-buy-agent}, \text{add-potential-customers}, \text{add-potential-sellers}, \text{agent-terminated}, \text{deal-made} \}$

$\mathbf{CP}a_5 = \{ \text{Call for proposal}, \text{Propose proposal}, \text{Reject proposal}, \text{Withdraw proposal}, \text{Accept proposal}, \text{Modify proposal}, \text{Acknowledge message}, \text{Terminate negotiation} \}$

$\mathbf{CP}a_6 = \{ \text{request-quotation}, \text{give-quotation}, \text{order}, \text{delivered}, \text{paid} \}$

$\mathbf{CP}a_7 = \{ \text{Accept Proposal}, \text{Agree}, \text{Cancel}, \text{Call for Proposal}, \text{Confirm}, \text{Disconfirm},$

Failure, Inform, Inform If, Inform Ref, Not Understood, Propagate, Propose, Proxy, Query If, Query Ref, Refuse, Reject Proposal, Request, Request When, Request Whenever, Subscribe }

$CPa_8 = \{Propose, Arrange, \mathbf{Request}, Inform, Query, Command, Inspect, Answer, Refine, Modify, Change, Bid, Send, Reply, Refuse, Explain, Confirm, Promise, Commit, \mathbf{Accept}, \mathbf{Reject}, Grant, Agree\}$

We first calculate the number of different communication links between them with $n = 8$, using Formula 2.

$$CL = (8^2 - 8) / 2 = 28$$

The resulting set of different communication links is given by:

$$DCL = \{ (a_1, a_2), (a_1, a_3), (a_1, a_4), (a_1, a_5), (a_1, a_6), (a_1, a_7), (a_1, a_8), (a_2, a_3), (a_2, a_4), (a_2, a_5), (a_2, a_6), (a_2, a_7), (a_2, a_8), (a_3, a_4), (a_3, a_5), (a_3, a_6), (a_3, a_7), (a_3, a_8), (a_4, a_5), (a_4, a_6), (a_4, a_7), (a_4, a_8), (a_5, a_6), (a_5, a_7), (a_5, a_8), (a_6, a_7), (a_6, a_8), (a_7, a_8) \}$$

Using Formula 4, we calculate the total number of communication primitives.

$$CPa_1 \cup CPa_2 \cup CPa_3 \cup CPa_4 \cup CPa_5 \cup CPa_6 \cup CPa_7 \cup CPa_8 =$$

{Initial_Offer, RFQ, Accept, Reject, Offer, Counter_Offer, CFP, Propose, Terminate, Acknowledge, Modify, Withdraw, Requests_Add, Authorize_Add, Require, Demand, Unable, Require-for, Insist_for, Demand_for, accept-offer, what-is-price, what-is-item, add-sell-agent, add-buy-agent, add-potential-customers, add-potential-sellers, agent-terminated, deal-made, Call for proposal, Propose proposal, Reject proposal, Withdraw proposal, Accept proposal, Modify proposal, Acknowledge message, Terminate negotiation, request-quotation, give-quotation, order, delivered, paid, Accept Proposal, Agree, Cancel, Call for Proposal, Confirm, Disconfirm, Failure, Inform, Inform If, Inform Ref, Not Understood, Propagate, Propose, Proxy, Query If, Query Ref, Refuse, Reject Proposal, Request, Request When, Request Whenever, Subscribe, Propose, Arrange, Query, Command, Inspect, Answer, Refine, Modify, Change, Bid, Send, Reply, Explain, Confirm, Promise, Commit, Grant, Agree}

$$CPT = |CPa_1 \cup CPa_2 \cup CPa_3 \cup CPa_4 \cup CPa_5 \cup CPa_6 \cup CPa_7 \cup CPa_8|$$

$$CPT = 82$$

Using Formula 5 we calculate the total number of different communication primitives.

$$|CPa_1 \cup CPa_2 \cup CPa_3 \cup CPa_4 \cup CPa_5 \cup CPa_6 \cup CPa_7 \cup CPa_8| = 82$$

$$|CPa_1 \cap CPa_2 \cap CPa_3 \cap CPa_4 \cap CPa_5 \cap CPa_6 \cap CPa_7 \cap CPa_8| = 0$$

$$DCPT = 82$$

We apply Formula 6 to calculate the level of heterogeneity for this set of heterogeneous agents.

$$\text{Level of heterogeneity} = 82 / 82 = 1$$

Although there are some agents which share some communication primitives among them, for example the subset of agents $\{ a_1, a_2, a_4, a_8 \}$ have in common the primitives $\{ Accept, Reject \}$; the level of heterogeneity among the eight agents, resulted in the maximal possible heterogeneity. Therefore we need to calculate the number of translations required in peer to peer communications for all the different communication links to evaluate the computational requirements for this set of heterogeneous agents. Using Formula 7 we calculate the sum of all required translations, for this case is equal to **599**.

Another important aspect that must be considered for a translation approach is the number of required individuals in the ontology. For this case the number of terms that must be defined in the ontology is equal to the total number of communication primitives.

$$|CPa_1 \cup CPa_2 \cup CPa_3 \cup CPa_4 \cup CPa_5 \cup CPa_6 \cup CPa_7 \cup CPa_8| = 82$$

5 Evaluation of Results

Figure 1 shows that more than the 70% of the 28 different communication links of the second case are totally heterogeneous, which represents a very difficult interoperability problem. To analyze if a translation approach would be a better solution we need to evaluate the number of required translations.

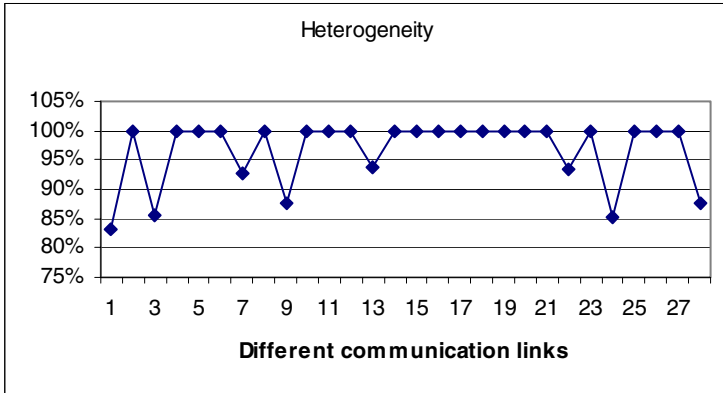


Fig. 1. Level of heterogeneity of the second case

Figure 2 shows the number of primitives and the number of different communication primitives of each different communication link. Figure 3 shows the number of required translations of each different communication link.

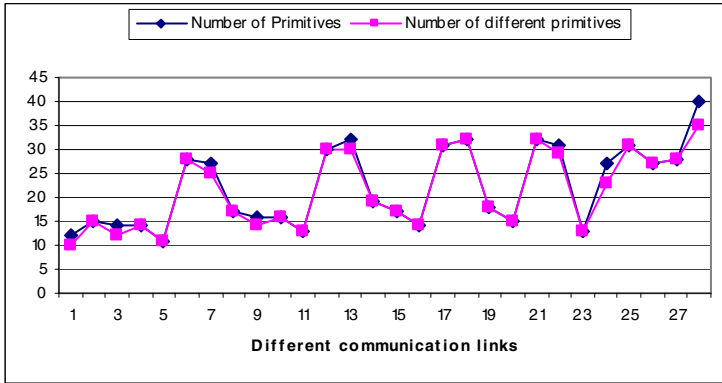


Fig. 2. Number of different communication primitives for the second case

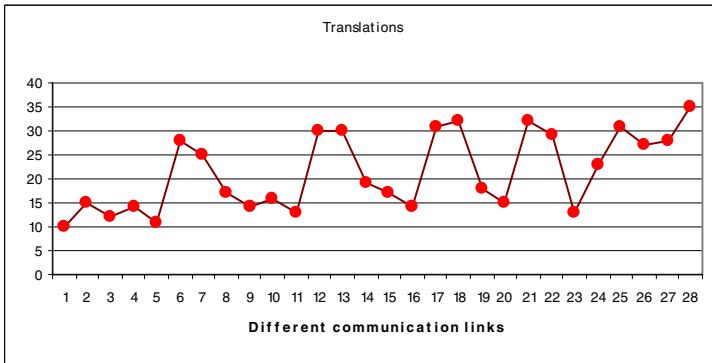


Fig. 3. Number of required translations of the second case

Comparing the three graphics we can see that there is a relation between the numbers of different communication primitives of Figure 2 with the number of required translations of Figure 3. However, there is not a direct relation between the level of heterogeneity and the number of required relations, as we would have supposed. Both cases are highly heterogeneous, the first case is 0.9 and the second is 1, which represents a minimal difference. The required number of individuals in the ontology for the first case is 20, while for the second is 82, this difference is significant. But the required number of translations has a bigger difference, the first case needs 36 translations while the second requires 599 translations.

6 Conclusions

In this paper we have presented a set of measures with the objective to evaluate the minimal computational requirements before selecting a solution for a given set of agents. Our measures are based on set theory, which has proven to be good representation formalism in other areas.

We showed how to use the set of measures with two sets of highly heterogeneous agents. We evaluated the results and obtained significant information for the developer of a solution. For example, many authors are proposing the use of ontologies to overcome heterogeneity. However, there are certain aspects that must be considered before implementing such a solution. An ontology will require to be populated with individuals, which will be the support of a translator. We need to measure the number of individuals in that ontology solution, and the number of required translations. For the second case, we can appreciate that although the number of individuals is low; the number of required translations is high. Therefore, we may infer that an increase in the number of participants will cause lower performance during communications.

To continue with this work we are extending our measures, considering a pragmatic approach to evaluate the differences and their impact in selecting a solution approach. There is also the need to evaluate communication scenarios in Web-based environments populated with more agents, and modeling the dynamics of such environments, where agents enter and leave communications any time.

References

1. Malucelli, A., Oliveira, E.: Towards to Similarity Identification to help in the Agents' Negotiation. In: Proceedings of 17th Brazilian Symposium on Artificial Intelligence, São Luis, Maranhão, Brazil (2004)
2. Pokraev, S., Reichert, M., Steen, M., Wieringa, R.: Semantic and Pragmatic Interoperability: A Model for Understanding. In: Proceedings of the Open Interoperability Workshop on Enterprise Modelling and Ontologies for Interoperability, Porto, Portugal (2005)
3. Rueda, S., García, A., Simari, G.: Argument-based Negotiation among BDI Agents. *Computer Science & Technology* 2(7) (2002)
4. Hübner, J.F.: Um Modelo de Reorganização de Sistemas Multiagentes. Ph.D. Dissertation, Universidade de São Paulo, Escola Politécnica (2003)
5. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 5(2), 199–220 (1993)
6. Willmott, S., Constantinescu, I., Calisti, M.: Multilingual Agents: Ontologies, Languages and Abstractions. In: Proceedings of the Workshop on Ontologies in Agent Systems, Fifth International Conference on Autonomous Agents, Montreal, Canada (2001)
7. de Souza, M.A.F., Hübner, J.F., Sichman, J.S., Varella Ferreira, M.A.: Interoperability in Multi-Agent Systems: Lessons Learned
8. Stuckenschmidt, H.: Exploring Partially Shared Ontologies for Multi-Agent Communication. In: Proceedings of the 6th International Workshop on Cooperative Information Agents VI (2002)
9. Boella, G., Damiano, R., Hulstijn, J., van der Torre, L.: A Common Ontology of Agent Communication Languages. *Applied Ontology* (2), 217–265 (2007)
10. Sosa, V.J., Bravo, M., Pérez, J., Díaz, A.: An Ontological Approach for Translating Messages in E-Negotiation Systems. In: Proceedings of the 7th International Conference on Electronic Commerce and Web Technologies EC-Web, Krakow, Poland (2006)
11. Bravo, M., Pérez, J., Velázquez, J., Sosa, V., Montes, A., López, M.: Design of a Shared Ontology Used for Translating Negotiation Primitives. *International Journal of Web and Grid Services* 2(3), 237–259 (2006)

ORM 2008 PC Co-chairs' Message

Following successful workshops held in Cyprus (2005), France (2006) and Portugal (2007), this was the fourth in a series of fact-oriented modeling workshops run in conjunction with the OTM conferences. Fact-oriented modeling is a conceptual approach to modeling and querying the information semantics of business domains in terms of the underlying facts of interest, where all facts and rules may be verbalized in language that is readily understandable by users working in those business domains. Unlike Entity-Relationship (ER) modeling and UML class diagrams, fact-oriented modeling treats all facts as relationships (unary, binary, ternary, etc.). How facts are grouped into structures (e.g., attribute-based entity types, classes, relation schemes, XML schemas) is considered a design level, implementation issue that is irrelevant to the capturing of essential business semantics. Avoiding attributes in the base model enhances semantic stability and populatability, as well as facilitating natural verbalization and thus more productive communication with all stakeholders. For information modeling, fact-oriented graphical notations are typically far more expressive than those provided by other notations. Fact-oriented textual languages are based on formal subsets of native languages, so are easier to understand by business people than technical languages like OCL. Fact-oriented modeling includes procedures for mapping to attribute-based structures, so may also be used to front-end other approaches. Though less well known than ER and object-oriented approaches, fact-oriented modeling has been used successfully in industry for over 30 years, and is taught in universities around the world. The fact-oriented modeling approach comprises a family of closely related dialects, the most well known being Object-Role Modeling (ORM), Cognition enhanced Natural language Information Analysis Method (CogNIAM) and Fully-Communication Oriented Information Modeling (FCO-IM). Though adopting a different graphical notation, the Object-oriented Systems Model (OSM) is a close relative, with its attribute-free philosophy. In December 2007, the Semantics of Business Vocabulary and Business Rules (SBVR) proposal was adopted by the Object Management Group, becoming the latest addition to the family of fact-oriented approaches.

Commercial tools supporting the fact-oriented approach include the ORM solution within Microsoft's Visio for Enterprise Architects, the CogNIAM tool Doctool, and the FCO-IM tool CaseTalk. Free ORM tools include InfoModeler and Infagon, as well as various academic prototypes. DogmaStudio is an ORM-based tool for specifying ontologies. NORMA, an open-source plug-in to Visual Studio, is currently under development to provide deep support for second generation ORM (ORM 2). Other ORM 2 tools under development include ActiveFacts and Richmond. Various SBVR tools are also currently under development. General information about fact-orientation and SBVR, respectively, may be found at www.ORMFoundation.org and http://omg.org/technology/documents/bms_spec_catalog.htm#SBVR.

This year we had 32 original proposals for workshop papers, with slightly fewer full paper submissions. After an extensive review process by a distinguished international program committee, with each paper receiving 3 or more reviews, we accepted the 15 papers that appear in these proceedings. Congratulations to the successful authors!

November 2008

Terry Halpin

A Metamodel for Enabling a Service Oriented Architecture

Baba Piprani¹, Chong Wang², and Keqing He²

¹ SICOM, Canada

² State Key Lab. of Software Engineering, Wuhan University, 430072, China
babap@attglobal.net,
wangchong_wuhu@yahoo.com.cn,
hekeqing@public.wh.hb.cn

Abstract. Process modelling initiatives generally develop their process models without much emphasis on data, burying their sequence of operations as a thread within a non-elementary process. More often than not, these buried operations are elementary atomic reusable components. The resulting models are generally not flexible or sufficiently reusable, suffering from update anomalies and redundancies. Addressing “service” as a major deliverable component, an ORM metamodel was developed in line with ISO 19763-5 Metamodel Framework for Interoperability: Metamodel for Process Model Registration, to harmonize atomic component processes using a control sequence and event models to enable the delivery of a totally flexible model set facilitating metamodel interoperability and cooperation between systems via their respective models. The paper provides a limited ORM based review of ISO 19763-5, and uses underlying component processes to develop a metamodel for a deliverable Services Oriented Architecture containing control sequence models, event models, and bridges to associated data models or web services.

Keywords: Services, Event Modelling, Service Oriented Architecture, ORM, ISO19763-5.

1 Introduction

Many businesses suffer from weak IT infrastructures consisting of disconnected databases, applications and services. This is even reflected in the glaring lack of documented business processes and their automatable counterparts in the form of IT Process Models.

A Conceptual Schema (as in ISO TR9007[1]) essentially reflects the static and dynamic rules of the enterprise. Processes address the dynamics and the behavior rules of an enterprise. Process modeling approaches have been around for decades in one form or another, each having their own syntax and semantics.

Process models, involving business processes, workflow, Web services etc., are deemed as a special kind of information resource along with complex structure, rich semantics and behavioral features.

International Standards Organization activities and several industrial consortia have contributed to standardization of domain specific process models using various representation notations and description languages for focused domains, such as BPMN[2] (Business Process Modeling Notation) for business process and OWL-s for Web services[3].

Most process modeling approaches concentrate on the flow of control for operations, weaving a complex scenario that may include several re-usable individual standalone processes in the form of a “service”. It is this inflexible set that is weak in its foundation and is not adaptable to change. Noting that the processes represent the ‘how’ of things to be addressed and dynamic behaviour in the enterprise, focus is lost on the ‘what’ of the enterprise, i.e. the business facts and semantics or the static behaviour of the enterprise.

Change in an enterprise essentially is reflected more in the ‘how’ part and much less so in the ‘what’ part i.e. there is more change that is reflected in how the business is done vs. less change on the facts themselves. Take for example the purchase of an airline ticket for a flight. The process has gone through a dramatic change from a manual paper ticket operation without computers, through issuing of paper tickets using computers, through e-tickets using computers, i.e. the ‘how’. But the facts that a person is travelling on a particular flight from an emplaning city to a deplaning city still remain the same.

2 The ISO 19763-5 Metamodel

In order to enable semantic interoperation between process models expressed in different modeling languages and promote further reuse based on them, ISO/IEC 19763-5[6][7] is introduced in this paper to provide a metamodel to register administrative structural information and meaningful semantics of process models, including workflows, business processes, Web services, software processes, etc. As an abstract facility, it focuses on the common structural and semantic content of process models expressed with different modeling languages, rather than their representations. Fig. 1 shows the overall structure of ISO/IEC 19763-5, i.e. Metamodel for process model registration(MPMR).

Concerning the construction of process models, Atomic_Process and Composite_Process are proposed to denote two kinds of process model. Atomic_Process is the simplest process model and corresponds to one-step execution. In contrast to Atomic_Process, Composite_Process comprises at least two sub-processes, which can be atomic processes or other composite processes. For either of them, we should designate the modeling language that the registered process model adopts and the purpose that should be achieved by Process_Modeling_Language and Goal respectively. Since process model can be identified as the transformation of input to output[8], it is obvious that one process model will have one or more Input to generate one or more Output as desirable products. If each input or output is taken as an information deliverer, then the involved objects or resources can be treated as corresponding information carriers. So in MPMR, all the objects, data and resources used in the process model can be instances of Artifact. Moreover, each artifact might play different roles specified by different communities in different cases. Therefore, artifacts respectively referred to the Input of one process and the Output of another process can be the same.

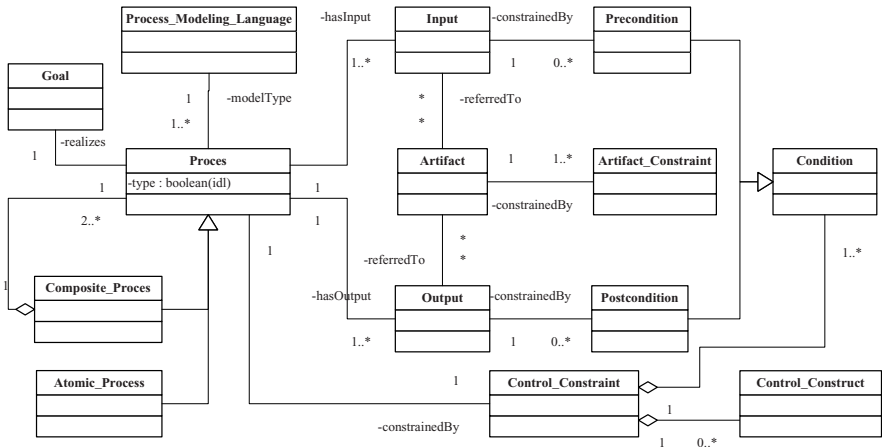


Fig. 1. Overall structure of Metamodel for process model registration (MPMR)

As for constraints between components within a certain process model, we define Artifact_Constraint to record relationship between Artifacts, which can be derived from knowledge base of domain or ontologies that artifacts are contained in, such as equivalence relation between two concepts. It also can be used to add semantics to referred resources and connect process models semantically or semi-automatically. Relatively, process is restricted with Control_Constraint. Particularly, due to the complexity of registered process models, two types of strategies are considered in MPMR. As for Atomic_Process, Condition is the only mandatory constraint, which has two subclasses, i.e. Precondition and Postcondition. Precondition is referred to Input to specify the information state that should be satisfied before execution, while Postcondition is restricted to Output to represent desirable outcomes when process is completed successfully. Considering Composite_Process, Control_Constraint becomes more complicated. It comprises Condition and Control_Construct because its sub-processes are connected with each other through at least one instance of Control_Construct. Specifically, Control_Construct here can be generalized as AnyOrder, Choice, Join, Split and Sequence. AnyOrder allows sub-processes to be executed in an unspecified order. Choice invokes one component of process model from a given collection. Join works when all of its components have been completed; Sequence means execution in order. Split produces at least two branches when the previous process model is executed successfully. Notice that inherent operation semantics of Control_Construct should be considered when specifying Precondition and Postcondition of Composite_Process.

Furthermore, Fig. 2 depicts the ORM Schema of ISO19763-5 metamodel as per published transforms from UML to ORM [9]. It is presented here to facilitate verification of the constructs by ISO.

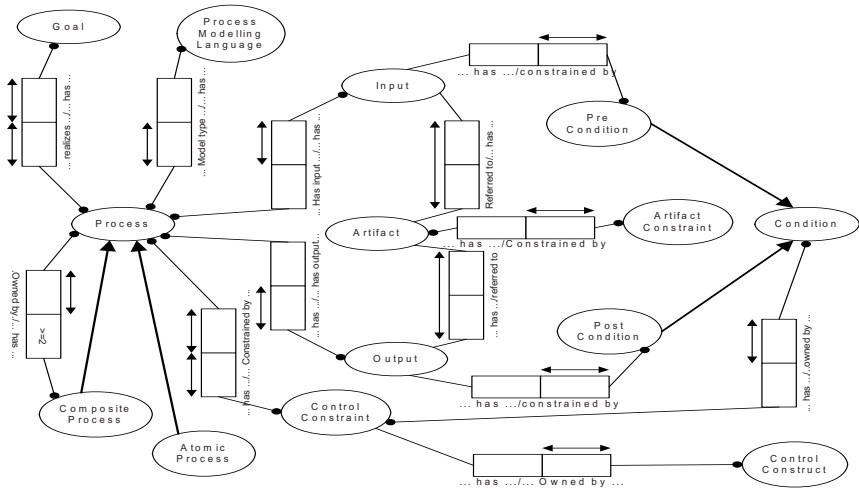


Fig. 2. ORM Schema of ISO 19763-5

3 Positioning the Process Model in SDLC vis-à-vis Services and Data

Business processes form the backbone of an enterprise in terms of its infrastructure to deliver services in association with proper supporting information. In Figure 3 we define a framework for positioning the various components of the involved infrastructure where each component environment is represented through some degree of one or more formalized models. Although ORM has been used as a candidate for modeling data semantics it is important to recognize that this framework also represents a generic model driven environment--including overlaps with OMG's Model Driven Architecture.

3.1 The Business Activity Model to the Semantic Modelling Phase

An initial forest-level picture pegging the boundaries of the set of stated requirements is first defined through a form of functional business decomposition to establish the overall scope. A business activity contributes to the achievement of an objective of the business. Each business activity in the hierarchy is decomposed until the lowest level activities (called elementary business activities or atomic processes that only handle a single unit of work and cannot be split further without loss of business meaning). This last (or lowest) level is denoted as level "n". The business decomposition stops at level "n-1", i.e. the level when the activity involves an "automatable part" and still maintaining a "business part", and where a further decomposition results in an automatable process involving primitive computer facilities of input / output.

The procedures for defining business activities and decomposition may be done differently by different people. Decompositions of the same business may be arrived at with different results based on which set of criteria is chosen, like business functions,

organizational etc.. This is quite acceptable and it does not matter, as long as all the business activities are being covered.

Why does this not matter? A business activity model is not a formal model i.e. there is not a formal grammar to support the business activity model. What matters is the data or information that is to be identified and formalized in the data usages of information flows from the lowest level process.

It is important that the lowest atomic processes represent a complete stand-alone, re-usable elementary task activity that cannot be split any further without losing meaning, and that these elementary tasks, while they may contain a processing sequence to accomplish that given elementary task, may not be connected or sequenced with other elementary tasks except for its own self to complete its given task---since this sequencing actually is a actually a service, and should be depicted by an independent stand-alone sequence model that controls the sequence of atomic processes.

A semantic data model is derived from the data usages in the information flows of these atomic processes. A semantic data model is a formal model with formal grammar associated with it, and is also known as a Computational Independent Model (CIM).

What this means, is that it does not matter how the business activities are organized, as long as the data usages have been recorded. No matter which alternate approaches of business activity modelling or decomposition is used---be it organizational based, product association based, business functionality based---the data usage information flows from the lowest level processes will ultimately result in “one” formal data grammar or semantic schema. This is because the final implementation is supported by a Services Model to achieve the business deliverables of the enterprise. The decomposition of business activities is only a means to achieve the formalization of the semantic data model required to support the enterprise information.

It is the Services Model that will bring the necessary atomic processes, their necessary sequences along with pre and post conditions and the Control Sequence Model to enable the carrying out of the necessary services for the enterprise as derived from the requirements. The Services Model is essentially driven by an Event Model which in its simplest incarnation depicts Time, i.e. Run the Backup Services at midnight every night, perform certain services at that start of a new year etc.

3.2 Bringing the Processes Together

Recall that the elementary business activity or an atomic process, while it may have its own internal sequence to complete the stated elementary task (e.g. change reservation date of hotel guest), it should not be associated with another process in any sequence except if that process is calling another elementary process to complete its given elementary task. For example the atomic process “change reservation date of hotel guest” will require a re-usable atomic process “select hotel guest folder” which will simply fetch the current reservation and other account details of the given hotel guest.

The sequence of processes to be performed is determined by a Services Model which has a set of processes that is driven by events and in turn uses a control sequence model that determines which process is to be performed for that particular service. Of course, the processes may require data from multiple database sources or URIs.

4 Processes in a Services Oriented Architecture (SOA)

The onset of a Service Oriented Architecture paradigm has resulted in a mixed bag of successes and failures. This is essentially due to the lack of any formalistic approaches being adopted towards the assemblage of processes involved in a service. The more intrinsically woven the atomic processes are, the more inflexible and less adaptable to change the service becomes.

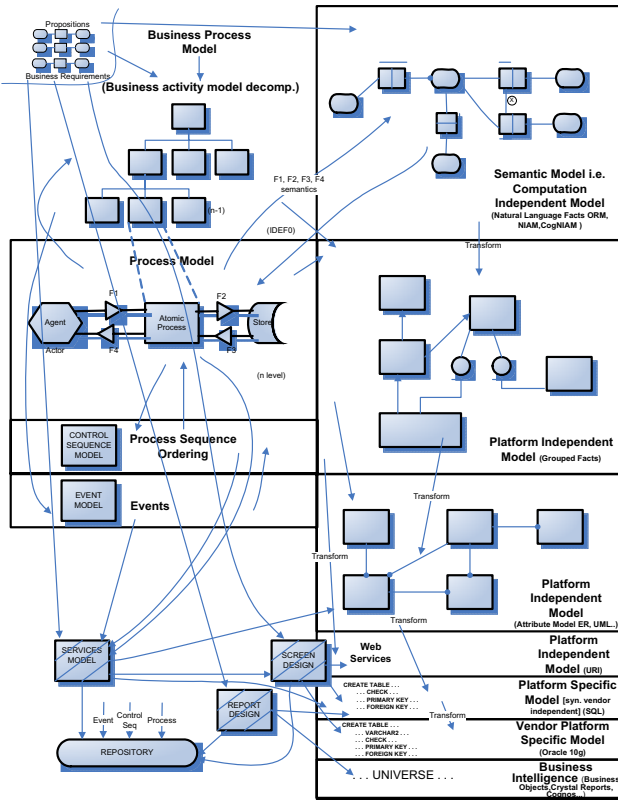


Fig. 3. Positioning the Services and Processes in the overall SDLC

The secret here is to divorce the control and sequence from within the service process to the level of ensuring that the contained atomic processes are identified, which as stand-alone elementary activities which only perform an elementary yet complete logical unit of work. For example, the Service Process of ‘register hotel guest’ can consist of (example set only---incomplete) ‘enter guest identification’, ‘assign room’ etc. The atomic process of ‘assign room’ simply checks to see if room is available under the desired parameters (no smoking, single room etc.) and assigns the room to the guest. The elementary task of room assignment needs to be completed as a whole, i.e. cannot half assign a room. This same ‘assign room’ atomic process

can be included in another Service Process of ‘Reassign Room’ where a registered guest is requesting a change in room. Note that this ‘Reassign Room’ service process does not need the ‘enter guest identification’ atomic process.

So here we have seen a de-coupling of complex processes into a Service Process’s constituent elementary or atomic processes.

5 ORM Schema of the Services Model

Extending the ISO 19763-5 metamodel to accommodate Services and Events, the resulting ORM Schema of the Services Model is presented in Fig. 4, which shows that an event may initiate one or more Services in a specified order. The initiation of the Service Step will have an Exception which in turn is an Event and may initiate Exception Processing. A Service may be involved in a hierarchy. A Service may be involved with the execution of one or more Atomic Processes in a specified sequence. The execution of an Atomic Process and Step will also have an Exception which in turn is an Event and may initiate Exception Processing.

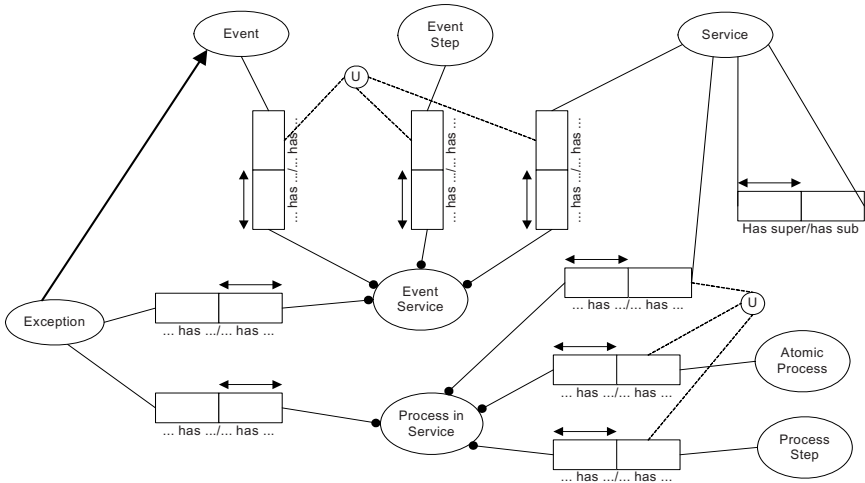


Fig. 4. ORM Schema of the Service Model

Fig.5 shows an ORM Schema that represents the Common Services Metadata. Each Service must have Service Metadata which may consist of Functionality Metadata, or Technical Metadata e.g. technical details, or Context metadata. In addition a Service must have one or more Service Providers which could be URIs or other agents including heterogeneous databases. A Service may be contained in a Service Group which may belong to a Service Category like Basic Services, Foundation Services, Management Services, Security Services, Business Services, and Identity Services etc. A Service Broker may access one or more Services Metadata.

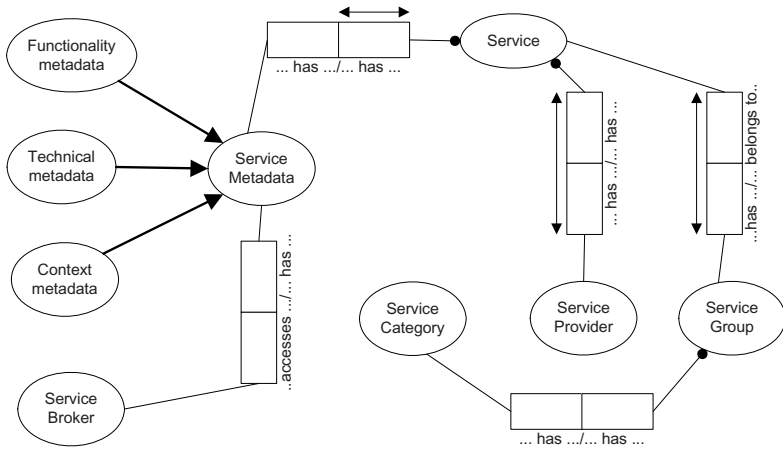


Fig. 5. ORM Schema of Common Services Metadata

6 A Strong SOA Overlay Based on Atomic Processes

We certainly want to avoid a spaghetti Services Oriented Architecture resulting from an ad hoc process of bringing together many interconnected and interwoven application systems. Recognizing that while Business Process Modelling is essentially a top-down process, the Services Oriented Architecture is a bottom-up process consisting of

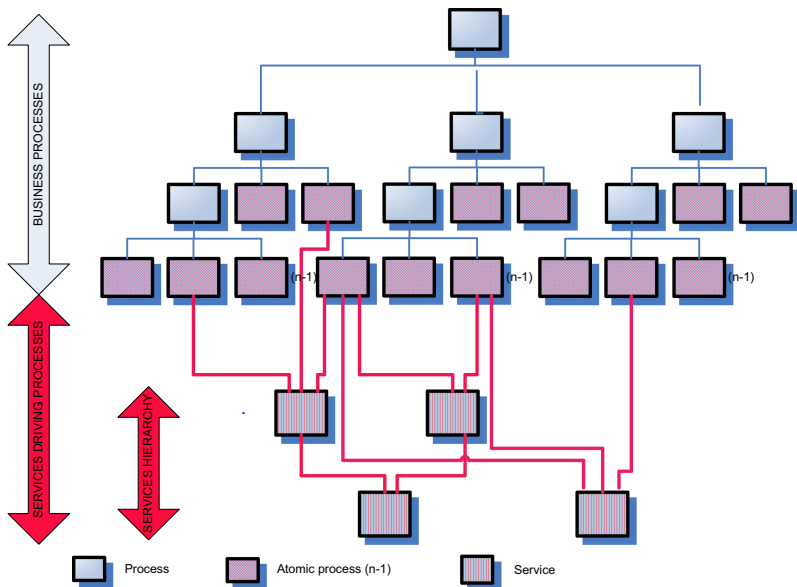


Fig. 6. Overlay Positioning Services and Processes

an assemblage of constituent atomic processes and/or services. It is important to recognize that while a service is a commitment of the business to achieving an outcome, the process is a mechanism to deliver or achieve that outcome.

Fig.6 shows the Services that may be members of a Service Group, positioned at the bottom half, which may execute one or more re-usable Atomic Processes as per the Event Services defined in Fig.4 and positioned in Fig.3.

7 Summary

Process and Service modelling initiatives generally develop their models without much emphasis on data. Moreover, process modelling paradigms generally bury process sequence of operations as a thread within a composite process. More often than not, these buried operations are elementary atomic reusable components. The resulting models are generally not flexible or sufficiently reusable, suffering from update anomalies and redundancies. Addressing “service” as a major deliverable component, an ORM metamodel was developed in line with ISO 19763-5 Metamodel for Interoperability: Metamodel for Process Model Registration, to harmonize atomic component processes using control sequence and event models to enable the delivery of a totally flexible model set facilitating metamodel interoperability and cooperation between systems via their respective models. The paper also provides a limited ORM based review of ISO 19763-5, and uses underlying component processes to develop a metamodel for a deliverable Services Oriented Architecture containing control sequence models, event models, and bridges to associated data models or web services.

Acknowledgement

The authors would like to acknowledge productive discussions with Dr. Robert Meersman, Dr. Sjr Nijssen, Paul Thompson, Dr. Yangfan He and Dr. Jian Wang.

This research project was supported by the National Basic Research Program of China (973) under Grant No.2006CB708302 and 2007CB310801, the National High Technology Research and Development Program of China (863) under Grant No.2006AA04Z156, the National Natural Science Foundation of China under Grant No.90604005, 60703018 and 60703009, and the Provincial Natural Science Foundation of Hubei Province under Grant No.2005ABA123, 2005ABA240 and 2006ABA228.

References

1. International Organization for Standardization(ISO): ISO Technical Report TR9007: Information processing systems—Concepts and Terminology for Conceptual Schema and the Information Base (1987)
2. Object Management Group (OMG): Business Process Modeling Notation (BPMN) Specification, Final Adopted Specification (2006)

3. Metcalf, C., Lewis, G.A.: Model Problems in Technologies for Interoperability: OWL Web Ontology Language for Services (OWL-S), CMU/SEI Technical Notes (CMU/SEI-2006-TN-018) (2006)
4. International Organization for Standardization (ISO): ISO/IEC 18629: Industrial automation systems and integration - Process Specification Languages (2004)
5. Morgan, T.: Business process modeling and ORM. In: Proceedings of On the Move to Meaningful Internet Systems: OTM 2007 Workshops, Vilamoura, Algarve, Portugal, pp. 581–590. Springer, Heidelberg (2007)
6. International Organization for Standardization (ISO): ISO/IEC 19763-5: Information technology – Framework for metamodel interoperability –Part 5: Metamodel for Process Model Registration, Working Draft (2008)
7. Wang, C., He, K.: Extending Metamodel Framework for Interoperability (MFI) to Register Networked Process Models. In: Dynamics of Continuous Discrete and Impulsive Systems-Series B- Applications & Algorithms, vol. 14(S6), pp. 72–78. Watam Press, Waterloo (2007)
8. International Organization for Standardization (ISO): 12207: Information Technology- Software Lifecycle Processes (1995)
9. Halpin, T.: UML data models from an ORM perspective: Parts 1-10. Journal of Conceptual Modeling, Inconcept (1998-2001), <http://www.orm.net>

Service-Oriented Conceptual Modeling

Peter Bollen

University of Maastricht
Faculty of Economics and Business Administration
Department of Organization & Strategy
P.O. Box 616
6200 MD Maastricht
The Netherlands
Tel: 31-43-3883715
Fax: 31-43-3884893
p.bollen@os.unimaas.nl

Abstract. Service-oriented computing (SOC) is a new paradigm that allows organizations to tailor their business processes, in such a way that efficiency and effectiveness goals will be achieved by outsourcing (parts of) business processes to web-based service-providers. In this paper we will show how semantic definitions of business process that are defined in an enterprise process base can be added to the regular list of application concept definitions in a fact-oriented conceptual modeling language.

Keywords: Service-oriented Computing, Service-oriented Architecture, SOC, SOA, ORM, CogNIAM, NIAM, Fact-orientation, Business Process Management, Conceptual Modeling.

1 Introduction

In the *service-oriented architecture* (SOA) paradigm, a *service requesting organization* (SRO) basically outsources one or more organizational activities or even complete business processes to one or more *service delivering organizations* (SDOs). The way this is done currently, is that the SRO ‘outsources’ a given business service to a ‘third-party’ SDO for a relative long period of time (3 months, a year). The selection and contracting activities are performed by organizational actors, i.e. managers responsible for the business processes in which the service(s) is (are) contained. Most of the current SDO’s provide ‘internet substitutes’ [1] for functions that used to be performed by an (integrated) SRO’s enterprise system, implying that the SRO’s that use these process services are shielded from the intrinsic complexities of these ‘substituted’ functionalities [2, 3].

Current approaches for web services, have limitations on a semantic and ontological level (among others) [4]. The problem with current approaches is that they cannot handle the semantic and ontological complexities caused by flexible participants having flexible cooperation processes. Semantic operability between participants (i.e. broker, SROs and SDOs) can only be achieved if the conceptual schema of the content, e.g. its ontology can be expressed totally and explicitly [5]. In most business organizations the

function that is responsible for information and knowledge management will have some kind of repository, schema or knowledge map that (ideally) defines the information objects (business repository or business ontology) and the semantic relationships between these business concepts (conceptual schema or a data description language (DDL) of some sort). At best (large) companies have a business glossary in which business concepts are defined precisely. When it comes to processes we must conclude that at best descriptions of procedural knowledge might be documented in some type of data flow diagram (DFD) or other process description logic (e.g. BPMN [6]). In most practical situations, however, the process logic is embedded in software code, and an explicit semantic description is lacking.

The application of the service-oriented paradigm that will lead to the most benefits for the SRO will be embedded in a semantic-web environment in which the ‘outsourcing’ decision in principle, can be made in real-time every time a service is requested [7]. This real-time level of decision making implies that the service-processes that are requested should be defined in such a way that the negotiation, contracting and execution of the service can take place in ‘run-time’ without ‘design time’ human intervention. In fact-oriented terminology we can say that a process is a fact-generating activity [8].

2 Related Work on Service-Oriented Architecture

In [9] a SOA (service oriented architecture) is provided. The basic elements from this service-oriented approach to distributed software design is given in figure 1.

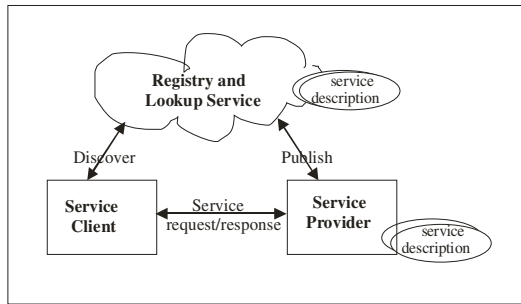


Fig. 1. SOA architecture as given in [9] and [10]

In the SOA from figure 1, service delivering organizations (SDOs) or service providers use the registry service (or broker [5] or service repository [11]) to publish their identity and a description of services that they provide. When a service requesting organization (SRO), service requestor [11] or service client, needs a service, it queries the lookup service (service discovery [12]) which will initiate the communication between SRO and SDO to establish a commitment regarding the service delivery [12]. We will take the SOA architecture from figure 1 as a basis for our further application of conceptual modeling on the SOA domain. We will thereby, explicitly distinguish three universes of discourse (UoD's): the client (SRO), the service provider (SDO) and the broker (or registry and look-up service).

3 Fact-Oriented Conceptual Modeling of the SRO

We will extend the current modeling capabilities of the fact-oriented approach with modeling constructs for the modeling of business services in the context of the service-oriented paradigm by extending the concepts definitions and derivation/exchange rule modeling constructs [13] to cater for ‘business services’ that can be provided by either the SRO itself or by one or more (external) SDO(s).

The commonly used process modeling approaches lack the capabilities to be used for this purpose [14]. In this section we will extend the fact-oriented conceptual modeling approach to cater for the definition of business functions (or parts of business processes) during design time in such a way that in a semantic web environment in which SRO and SDO’s can interchange their domain ontologies and thereby in runtime can decide which of the relevant SDO’s will be partner to deliver the requested service for a given business transaction.

We will present the elements from the fact-oriented knowledge reference model (KRM) and see how they can be applied in the situation in which SDOs are involved in (interorganizational) business processes. We will use as a running example for the UoD of the SRO, the (fictitious) ABC company, and focus on the carrier selection process for customer shipments.

3.1 The SRO UoD: The ABC Company’s Carrier Selection Business Process

ABC is a business that operates a number of ‘brick-and-mortar’ stores. Although the company does have an internet retail-website, it sometimes receives order request for deliveries via mail, e-mail or fax, outside the sales region it serves and in some cases even outside the country it operates in, and sometimes it receives ‘overseas’ order requests. Especially for the latter order category, ABC can make an additional profit by shipping the order using the cheapest carrier at any given point in time. The shipping fee, they charge to their customer is a constant fee. The customer has the choice between standard shipping and express shipping. The ABC company, has a logistics department in which 1 person is responsible for the shipment of continental and overseas orders. Since this person, has also other logistics responsibilities, he/she can not afford to spend too much time trying to search for the best transportation deals. It might be beneficial for ABC, to ‘outsource’ the carrier selection process to a third-party, in this case a service delivery organization (SDO).

In the next sub-sections of this section we will apply the Knowledge Reference Model (KRM) [15] on the example UoD of the ABC carrier selection business process.

3.2 Knowledge Domain Sentences

If we apply the KRM on the ABC case study we can conclude that the fact types that concern the customer, order, destination, shipment conditions and carrier do not change in the new ‘service-oriented’ situation. On the other hand the design-time specification of SDO selection, can be considered a new UoD, in which following (types of) domain sentence is relevant (amongst others):

'The order with ordercode 23456 of service requesting organization having organization code 34567 has a cargo dimension having a size for which the width is 3 meters, the length is 1 meter and the height is 2 meters'.

3.3 Concept Definitions and Naming Conventions for Concepts Used in Domain Sentences

We will now take this set of 'explicit' verbalizations and abstract them into a set of concept definitions and fact type readings in a fact type diagram. This list of structured concept definitions, should facilitate the comprehension of knowledge domain sentences and comprise the business domain ontology [16].

Table 1. List of concept definitions for SRO (not complete)

Concept	Definition
SRO	An [organization] that potentially can request a service from a third party organization.
SDO	An [organization] that delivers a service to a [SRO]
Cargo	A product shipment from a [SRO] to a customer
Dimension	Size of [cargo] as length* width * height
Dimension code	A name from the <i>dimension code</i> name class that can be used to identify a [dimension] among the set of [dimension]s
Size	Depicts the extent in meters of any of the three elements of a [dimension]
# of meters	A name from the <i>two decimal number</i> name class that can be used to identify a [size] among the set of [size]s
Volume	Depicts the extent in cubic meters of a three-[dimension]-al package
# of cubic meters	A name from the <i>two-decimal number</i> name class that can be used to identify a [dimension] among the set of [dimension]s
Delivery type	A generally agreed upon type of delivery by a [service requesting organization] and a service registry organization or broker that is characterized by a maximum [dimension]
Delivery type code	A name from the delivery type code name class that can be used to identify a [delivery type] among the set of [delivery type]s.
Contract base	Type of commitment between [service delivery organization] and [SRO]
Contract base code	A name from the <i>contract base code</i> name class that can be used to identify a [contract base] among the set of [contract base]s.
'Per transaction' contract base	A specific value for a [contract base code] that means that a contract between a [SDO] and a [SRO] change per transaction on the discretion of a [SRO].
'Weekly renewal' contract base	A specific value for a [contract base code] that means that a contract between a [SDO] and a [SRO] can change per week on the discretion of a [SRO].
Order	A request to ship a package to a customer
Order code	A name from the <i>order code</i> name class that can be used to identify an [order] among the set of [order]s
Carrier	A third party logistics organization that ships packages for an [order] from a [SRO] to a client of the [SRO]
Carrier code	A name from the <i>carrier code</i> name class that can be used to identify a [carrier] among the set of [carriers]s that exist in the world.

3.4 Fact Types and Fact Type Readings

The domain sentences from the former sections can be abstracted and will lead to fact types and associated fact type readings in figure 2. These fact types can be used as a starting point for a further explicitation and encoding of business rules in terms of constraints on the allowed populations of the fact type diagram in figure 2.

3.5 Population State (Transition) Constraints for the Knowledge Domain

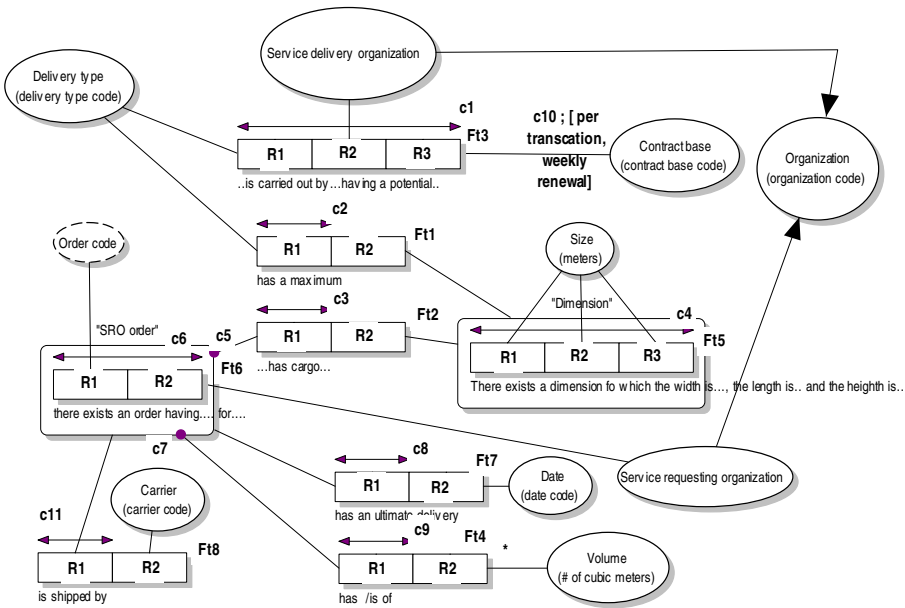
In case a standard between the SDO's and SRO's and service broker has been implemented, in which it is agreed upon that: *for any (predefined) delivery type at most one maximum dimension can exist*, we can show this as a uniqueness constraint of fact type Ft1 that covers the role R1. A further formalization of the allowed communication within the SRO's UoD's is the convention that *a given order by a given SRO must have exactly one dimension*. The latter business rule is encoded in the fact type model in figure 2 as a uniqueness constraint spanning role R1 of fact type Ft2 in combination with a mandatory role on the nested entity type SRO-order.

3.6 Derivation Rules, Exchange- and Event Rules

In addition to the business rules that can be expressed as population state (transition) constraints, we can add business rules that can derive 'new' fact instances from 'old' fact instances. An example of such a derivation rule can be applied for fact type Ft4. We assume that *a volume is the multiplication of the three dimensions* figures that are modeled in fact type Ft2/Ft5. This derivation rule can be modeled as derivation rule *dr1* in figure 2 in which formula: $Ft4.R2 = Ft5.r1 * Ft5.r2 * Ft5.r3$ is contained. We note that in a service oriented architecture, derivation rules play an important role (next to exchange rules, see next section) because SRO's 'outsource' the execution and management of these rules to SDO's. It's therefore, paramount to incorporate the semantic definition of these derivation- and exchange processes into the list of concept definitions. The addition to the list of concept definitions together with the fact type diagram and the business rules that are defined on the fact type structure are given in figure 2.

The last two elements in the KRM are the exchange rules and the event rules. For those fact types for which no derivation rules are given, we can in principle define an exchange process that states that a fact can be added or removed to/from the information base unconditionally *or* under some condition [17]. In the latter case we will give an event rule that specifies under what condition the exchange (addition or deletion) of a fact instance takes place. In case a number of instances of one or more fact types will be added and/or deleted under (possibly different) conditions on the information base in one or more event rules, it is recommended to add the definition of such a 'transaction' into the list of concept definitions. In our example, the process *calculate volume* is implemented within the sphere of influence of the organization itself. The process is made explicit in the form of derivation rule: *Define order has Volume (cubic meters)*, that is listed at the bottom of figure 2. The process determine carrier for order, however is outsourced to some SDO. We remark, that the definition of the process (or service description) as an imperative, in the case that SDO's who provide such a web-service are selected in run-time on a per transaction base.

Process: Calculate Volume	A process that has a result: a rough indicator of the cubic [volume] of a package which is determined by multiplying its width, height and length. <Create(s) instance(s) of Ft4>
Process: Add order	A transaction in which the [order] and the [dimension] and [delivery date] of the [order] are added to the information system. <Create(s) instance(s) of Ft2 and Ft7>
Process: Determine carrier for order	This process leads to the selection of a specific [SDO] for the shipment of an [order] under the best possible conditions for [delivery time] and [shipment price] <Create(s) instance(s) of Ft8>



Define Order has Volume (cubic meters)

As Order has cargo Dimension **and** There exist a dimension for which the width is Size₁ and the length is Size₂ and the height is Size₃ **and** Volume= Size₁ * Size₂ * Size₃

Fig. 2. Complete conceptual schema for SRO (in combination with table 1)

Adding the semantic definition of a (business) process to the list of concept definitions, is a pragmatic extension of the current definition of the list of definitions, which normally contains definitions for concepts in the ontology. From a theoretically point of view, however, if we consider a(n) (enterprise) process base [18, 19] as part of our UoD, then a semantic definition of a process type, should per definition be contained in the list of concept definitions.

4 Fact-Oriented Conceptual Modeling of the SDO

In this section we will look at the Universe of Discourse of a web-service that provides carrier selection services for SRO's. One of the main processes within this UoD's is the up-to-date acquisition of carrier data regarding latest offers, in terms of shipment conditions, and prices for each delivery type and possibly delivery (sub)-types depending upon each individual carrier. This web-service organization has as objective to match SRO's with carriers normally for a small fee per transaction. We will see that ontological commitments need to be established between SRO's and SDO's on a 'design'-time level. This means that key concepts for web-based service transactions will be harmonized (as can be checked for example in the list of concept definitions in tables 1 and 2, for the concept *delivery type* and *carrier*). On the other hand, promotional concepts and other rating schemes can be introduced on the fly, at any time by a carrier. For many of these promotional campaigns and or new tariff schemes, it will not be feasible to establish ontology harmonization between the SDO and these carriers at all times. To cater for this, we need modeling constructs that allow us to deal with the runtime changes in domain concepts as used by SDO's in their carrier selection processes on behalf of their SRO customers. We will show now in our example list of definitions and conceptual schema for the UoD of the SDO can be modeled for these short-term runtime definitions of domain concepts. We note that a 'snapshot' of delivery types for every carrier that that is considered by a carrier-selection SDO will be modeled as a populations of fact type Ft1 in the conceptual

Table 2. List of concept definitions for SDO (not complete)

Concept	Definition
Local delivery type	A label to refer to a specific type of service provided by a specific [carrier]
Carrier delivery type	A [local delivery type] that is offered by a [carrier]
Promotional price	A price that is charged per kg for a delivery service during a number of [week]s in a promotional period
Maximum dimension	The maximum [size] for length * the maximum [size] for width * the maximum [size] for height of an [order] for which a given [delivery type] is still valid
Maximum delivery period	The maximum value for [Period length in days] it takes to deliver a package to a client of a [SRO]
Process: Classify service offering	A process that has a result a classification for a [local delivery type] offered by a [carrier] in terms of an instance [delivery type] that has been defined by a [SRO] and [SDO]. <Create(s) instance(s) of Ft108>
Process: Add service offering delivery length	A process that has a result that a maximum delivery length for a [carrier delivery type] is entered into the information base. <Create(s) instance(s) of Ft106>
Process: Add service offering standard price	A process that has a result that a [standard price] for a [carrier delivery type] is entered into the information base <Create(s) instance(s) of Ft101>
Process: Add service offering promotional price	A process that has a result that a [promotional price] during one or more [weeks] for a [carrier delivery type] is entered into the information base <Create(s) instance(s) of Ft102 and Ft107>

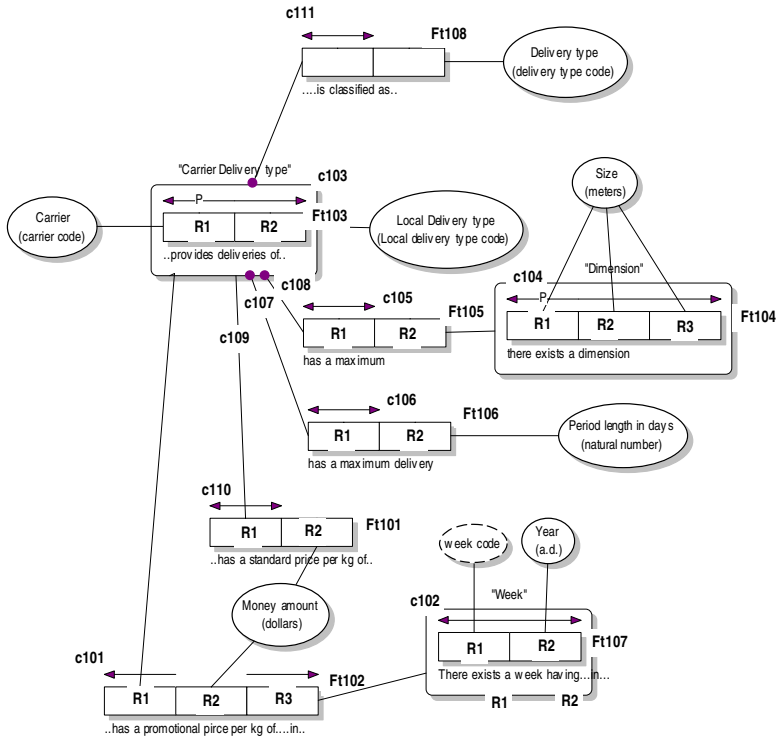


Fig. 3. Conceptual schema for SDO (in combination with table 2)

schema of the carrier selection SDO in figure 4. The domain sentences that will be communicated will contain (amongst others) the following expressions:

‘The carrier having carrier code *DHL* provides deliveries of local delivery type with local delivery type code *DHL express*. This carrier delivery type has a maximum dimension having a size for which the width is 3 meters, the length is 3 meter and the height is 3 meters’ .

‘The carrier having carrier code *DHL* provides deliveries of local delivery type with local delivery type code *DHL express*. This carrier delivery type has a maximum delivery period length in days of *14*’ .

We now see that the carrier selection broker service not only provides the best deal for a service requesting organization, but also performs the role of ‘ontological harmonizer’ between the SRO’s and the carriers by introducing and defining the concepts of *local delivery type* and *carrier delivery type*.

In table 2 we have provided the extended list of concept definitions for this example UoD of a service delivery organization in which the definitions of the fact generating processes are incorporated. In figure 3 we have given the complete conceptual schema for the example carrier selection process within the UoD of the SDO.

When we inspect the conceptual schema for our example SDO in figure 4, we can say that one of the 'core business processes' for the SDO, is establishing ontological harmonization in 'run-time'. This means that the SDO will populate fact type Ft108 in figure 3, by continuously scanning for recent service offerings provided by existing and new carriers. This business process mainly scans and interprets these service offerings, and as a result will 'label' these offerings and subsequently classify them, in the terminology, that was established between the SRO's and SDO's, via a broker or registry service. In the running example of this article, we have limited ourselves to only depict a few relevant fact types that will be used in practice. In a real-life conceptual schema a multitude of this number of fact types might actually be used in the communication, between SDO, SRO and registry service.

5 Conclusions

In this article we have applied business modeling concepts in ORM/CogNIAM to cater for the explicit modeling of a application domain's ontology. This allowed us to capture the definitions of the fact-generating business processes and incorporate them into the list of concept definitions. Such a conceptual schema will allow us to communicate the definition of business processes with potential external agents, e.g. customers, suppliers, web-service brokers, whose identity is not yet known to us at design time.

In line with semantic web developments, the conceptual schema needs a communication part that contains 'definition' instances to be shared with the potential agents in order for them to be able to communicate effectively and efficiently with a ('web-based') business application in which the 'traditional' allowed communication patterns and their state (transition) constraints will not be violated.

In this article we have precisely shown how such a communication part can be established for the business processes that are defined in the enterprise process base. By using a conceptual model that contains the explicit semantic definitions of those enterprise processes, the quality and ease-of-use of the target (web-based) application will be increased significantly.

Another advantage of applying ORM/CogNIAM for capturing an application or a (relatively complex) domain's ontology is in its flexibility to use it even to model communication between agents in which (explicit) ontological harmonization at a type or schema level is not possible or desirable. By adding 'run-time' concepts as populations of (typed) concepts for which an ontological harmonization already has been established.

References

1. Siau, K., Tian, Y.: Supply chains integration: architecture and enabling technologies. *Journal of Computer Information Systems* 45, 67–72 (2004)
2. Estrem, A.: An evaluation framework for deploying web services in the next generation manufacturing enterprises. *Robotics and Computer Integrated Manufacturing* 19, 509–519 (2003)

3. Baina, K., Benali, K., Godart, C.: Discobole: a service architecture for interconnecting workflow processes. *Computers in Industry*, 57 (2006)
4. Shen, W., et al.: An agent-based service-oriented integration architecture for collaborative intelligent manufacturing. *Robotics and Computer Integrated Manufacturing* 23, 315–325 (2007)
5. Yue, P., et al.: Semantics-based automatic composition of geospatial web service chains. *Computers & Geosciences* 33, 649–665 (2007)
6. OMG, Business process modelling notation (BPMN) specification. OMG (2007)
7. Menascé, D., Ruan, H., Gomaa, H.: QoS management in service-oriented architectures. *Performance Evaluation* 64, 646–663 (2007)
8. Bollen, P.: Conceptual process configurations in enterprise knowledge management systems. In: *Applied computing 2006*, ACM, Dijon (2006)
9. McIntosh, R.: Open-source tools for distributed device control within a service-oriented architecture. *Journal of the Association for Laboratory Automation* (9), 404–410 (2004)
10. Jardim-Goncalves, R., Grilo, A., Steiger-Garcia, A.: Challenging the interoperability between computers in industry with MDA and SOA. *Computers in Industry* 57, 679–689 (2006)
11. Mokhtar, S.B., et al.: Easy: efficient semantic service discovery in pervasive computing environments with QoS and context support. *The Journal of Systems and Software* (2007)
12. Cotroneo, D., et al.: Securing services in nomadic computing environments. *Information and Software Technology* (2007)
13. Bollen, P.: Fact-oriented modeling in the data-, process- and event perspectives. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM-WS 2007, Part I. LNCS*, vol. 4805, pp. 591–602. Springer, Heidelberg (2007)
14. Morgan, T.: Business Process Modeling and ORM. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM-WS 2007, Part I. LNCS*, vol. 4805, pp. 581–590. Springer, Heidelberg (2007)
15. Nijssen, G., Bijlsma, R.: A conceptual structure of knowledge as a basis for instructional designs. In: *The 6th IEEE international conference on Advanced Learning Technologies, ICALT 2006*, Kerkrade, The Netherlands (2006)
16. Bollen, P.: Extending the ORM conceptual schema design procedure with the capturing of the domain ontology. In: *EMMSAD 2007*, Tapir Academic Press, Trondheim (2007)
17. Bollen, P.: Using Fact-Oriented for Instructional Design. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops. LNCS*, vol. 4278, pp. 1231–1241. Springer, Heidelberg (2006)
18. Nijssen, G.: An axiom and architecture for information systems. In: *Information systems concepts: an in-depth analysis* (1989)
19. Bollen, P.: Fact-Oriented Business Service Modeling. In: *12th international workshop on Exploring Modeling Methods in Systems Analysis and Design (EMSSAD 2007)*, Trondheim, Norway (2007)

Temporal Modeling and ORM

Terry Halpin

Neumont University, Utah, USA
terry@neumont.edu

Abstract. One difficult task in information modeling is to adequately address the impact of time. This paper briefly reviews some popular approaches for modeling temporal data and operations, then provides a conceptual framework for classifying temporal information, and proposes data model patterns to address time-impacted tasks such as modeling histories, and tracking entities across time as they migrate between roles. Special attention is given to capturing the relevant business rules. While the data modeling discussion focuses on Object-Role Modeling (ORM), many of the basic principles discussed can be adapted to other approaches such as Entity Relationship Modeling (ER) and the Unified Modeling Language (UML).

1 Introduction

One challenging aspect of information modeling is to deal appropriately with temporal data and operations. This paper discusses how various temporal issues can be addressed at the conceptual level. The treatment focuses on Object-Role Modeling (ORM), a fact-oriented approach for modeling, transforming, and querying information in terms of the underlying facts of interest, where facts and rules may be verbalized in language readily understandable by non-technical users of the business domain. However, much of the discussion can be adapted to other data modeling approaches such as Entity Relationship Modeling (ER) [5] and the class diagramming technique within the Unified Modeling Language (UML) [19].

Unlike ER modeling and UML class diagrams, ORM models are attribute-free, treating all facts as relationships (unary, binary, ternary etc.). ORM includes procedures for mapping to attribute-based structures, such as those of ER or UML. In addition to ORM, fact-oriented modeling includes a number of closely related approaches, such as Natural language Information Analysis Method (NIAM) [29] and Fully-Communication Oriented Information Modeling (FCO-IM) [1]. For a basic introduction to ORM see [13], and for a thorough treatment see [15]. For a comparison of ORM with UML see [11].

Business rules include constraints and derivation rules. Static rules apply to each state of the information system that models the business domain (e.g. each person was born on at most one date). Dynamic rules reference at least two states, which may be either successive (e.g. no employee may be demoted in rank) or separated by some period (e.g. invoices ought to be paid within 30 days of being issued). While ORM provides richer graphic support for static rules than ER or UML provide, ORM as yet cannot match UML's support for dynamic rules.

Since the 1980s, many extensions to fact-orientation have been proposed to model temporal aspects and processes. The TOP model [10] allows fact types to be qualified by a temporal dimension and granularity. TRIDL [4] includes time operators and action semantics, but not dynamic constraints. LISA-D [18] supports basic updates. Task structures and task transactions model various processes [17], with formal grounding in process algebra. EVORM [24] formalizes first and second order evolution of information systems. Explorations have been made to address reaction rules [e.g. 16], and a proposal has been made to extend ORM with a high level textual language to specify dynamic rules in a purely declarative fashion [3].

Some fact-based approaches that share similarities with ORM include support for modeling temporality or system dynamics. For example, one extended fact-based model caters for different calendric systems and temporal operators [23], T-ORM provides basic support for temporal object evolution [8], the CRL language in TEMPORA enables various constraints, derivations and actions to be formulated on Entity-Relationship-Time (ERT) models [26, 28], and the OSM method includes both graphical and textual specification of state nets and object interactions [9].

Attribute-based methods such as UML and some extensions of ER incorporate dynamic modeling via diagrams (e.g. UML state charts and activity diagrams), with recent approaches such as the Business Process Modeling Notation (BPMN) gaining popularity for workflow modeling. The MADS (Modeling of Application Data with Spatio-temporal features) approach [22] extends ER with deep support for temporal data types and operators. For textual specification of dynamic rules, the most popular approach is the Object Constraint Language (OCL) [21], but the OCL syntax is often too mathematical for validation by nontechnical domain experts.

The rest of this paper is structured as follows. Section 2 reviews some standards and proposals for modeling temporal data and operations. Section 3 discusses conceptual issues and patterns for modeling facts that include temporal information. Section 4 proposes data model patterns, including dynamic rules where necessary, to capture histories of entities as they migrate between roles. Section 5 summarizes the main results, and lists references.

2 Temporal Data Standards and Proposals

Industrial standards and proposals for temporal data typically identify three main temporal data types: instant; duration; and period. An *instant* is a point in time (e.g. 2008 July 4, 2:00 p.m. MDT). A *duration* is a length of time (e.g. 2 weeks): this term is used in both ISO 8601 [19] and XML schema (www.w3.org/TR/xmlschema11-2/), but is called “interval” in the SQL standard (www.iso.org). A *period* is an anchored duration of time (e.g. 2008 July 4 ... 2008 July 7 PST). This term is used in SQL/Temporal (currently on hold), but is called *time interval* in ISO 8601 and *interval* in OWL-Time (www.w3.org/TR/owl-time/). Hence the term “interval” needs to be used with care. In ISO 8601, periods are closed (they have both a start and an end), but in OWL-Time they may be open (e.g. today onwards). In OWL-Time, a period with nonzero extent (if closed, it ends after it starts) is a *proper* period.

Most temporal standards draw from ISO 8601, which specifies many temporal terms, calendric systems (e.g. Gregorian, Julian), time zones (e.g. UTC, MDT), date

and time formats etc. The SQL standard includes basic support for date, time, date-time and “interval” (in the sense of duration). XML Schema supports these and several other temporal data types (e.g. gYear for Gregorian year). The Time Markup Language (TimeML) covers date/time concepts as well as linguistic expressions to describe events (www.timeml.org/site/index.html). OWL-Time, a working draft to extend the Web Ontology Language OWL (www.w3.org/2004/owl) includes a set of temporal classes and predicates, and logical axioms about these. A subgroup of the Object management Group (OMG) is currently working to provide a unified treatment of basic temporal concepts for use in multiple approaches, including SBVR (Semantics of Business Vocabulary and Business Rules).

Instants are strictly ordered on a time axis, and may be compared using temporal operators such as $<$ (for “is before”) and \leq (for “is at or before”).

But many different proposals exist for an appropriate set of *temporal operators between periods* (time intervals). Many make use of *Allen’s operators* [1], although some of these proposals [e.g. 7] wrongly construe many of Allen’s definitions. Fig. 1 visually depicts Allen’s operators as 13 mutually exclusive relationships between an ordered pair of closed, proper periods P_1 and P_2 .

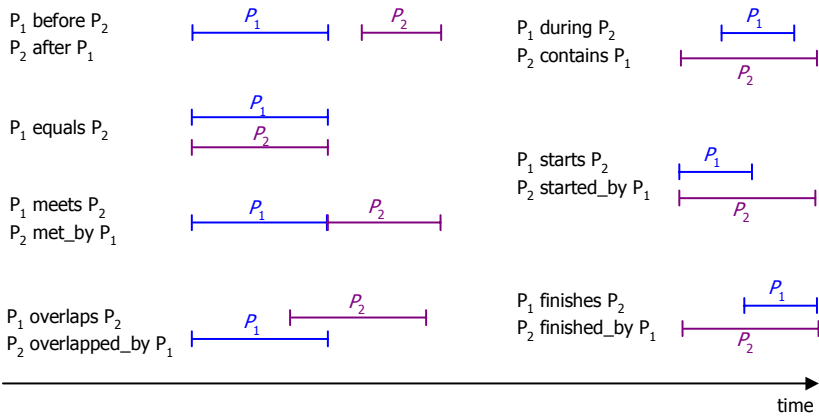


Fig. 1. Allen’s operators for comparing closed, proper periods

OWL-Time accurately adopts Allen’s operators, but this is unfortunate, as these operators are often poorly named, poorly defined, or poorly chosen. To begin with, overlaps and meets are intuitively understood as symmetric, but here are made asymmetric. If such asymmetric operators are to be used, they should be given intuitive names (e.g. leftOverlaps, rightOverlaps, leftMeets, rightMeets). Moreover, using our intuitive understanding of the terms, the contains, starts and finishes operators are too restrictive (e.g. equals should be treated as a special case of these, and starts should be a special case of during etc.).

Allen’s before and after operators between periods require that the end of $P_1 <$ the start of P_2 . This could be acceptable if we adopt a quantized view of time, where time is composed of atomic *chronons*, since that would allow the periods to be contiguous. However if we assume that time is continuous, this would not allow the periods to be

contiguous, which goes against our common sense notion of “before”. For example, one would normally agree that the Jurassic period is before the Trassic period (it immediately preceded it) and that yesterday is before today. Since OWL-Time and many other approaches leave the question open whether time is discrete or continuous, this choice of “before” is bound to confuse.

If time is regarded to be continuous, it is better to define “before” between periods so that the end of $P_1 \leq$ the start of P_2 . With this definition, it follows that yesterday is before today, even if we adopt the ISO 8601 definition of calendar day as a “time interval starting at midnight and ending at the next midnight, the latter being also the starting instant of the next calendar day” [698]. Note that with this definition, any given midnight occurs on exactly two calendar days!

With this definition of calendar day, and using Allen’s operators, yesterday meets today (yesterday’s end is today’s start) but does not overlap with today (since Allen’s overlap requires yesterday’s end to precede today’s start). It seems preferable to define periods to *overlap* if and only if they have an instant in common. Apart from being symmetric, this is consistent with the way overlaps is defined in set theory and mereology. As illustrated later, it is also useful to distinguish between *trivial overlap* (where periods have exactly one instant in common) and *nontrivial overlap*.

Allen’s *meets* operator is flawed, not only in being asymmetric, but in failing to cater for discrete time. If time is discrete, we should define periods to meet if they are contiguous (the end chronon of one immediately precedes the start chronon of the other). Note that this is one way to avoid the temporal version of the classic problem about where the midpoint goes when a line is divided in two [698, p. 110].

Whether or not time is continuous, we can measure time only to a limited accuracy, which effectively makes it discrete for information modeling purposes. Moreover, when recording information, we often choose a coarser temporal granularity than is physically attainable (e.g. we might track a patient’s blood pressure at most daily or hourly). Pragmatically, we often juxtapose periods of a coarse granularity when tracking history rather than treating the end of one period to be the start of the next. For example, when updating an employee’s salary, the new salary period is typically set to one day after the previous salary period. This avoids problems such as assigning two different salaries to an employee at the instant his/her salary is updated.

Another problem with Allen’s operators is that they are often of little use pragmatically. In practice, one often needs instead to apply constraints involving our intuitive notions or overlapping, nonoverlapping, containment, etc.

Recently we investigated OWL-Time from an ORM perspective, and found it to be seriously deficient. Apart from its unwise adoption of Allen’s operators, OWL-Time’s axiomatic development appears to be problematic (partly because of its silence on the discrete/continuous time issue), and is incomplete. As a simple example of the latter, Fig. 2 shows an ORM schema for a fragment of the OWL metaschema dealing with duration descriptions. The inclusive-or constraint (circled dot) and preferred external uniqueness constraint (circled double-bar), are not captured in OWL-Time, but are clearly needed. As a general comment about OWL itself, OWL models are much easier to formulate if generated from ORM rather than working directly in OWL.

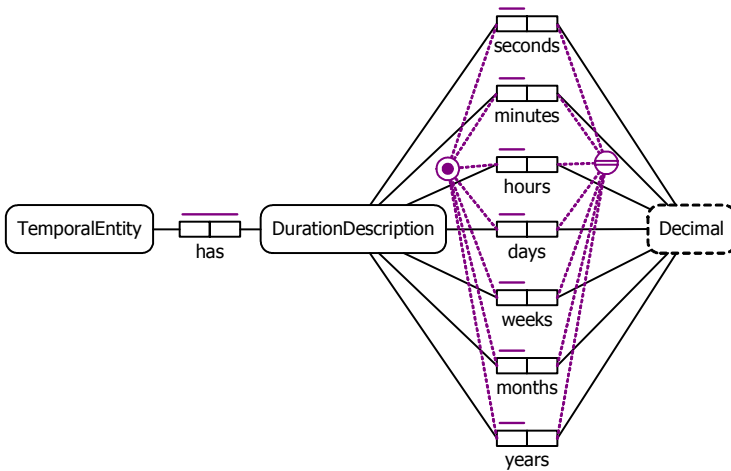


Fig. 2. ORM schema for duration descriptions in OWL-Time (constraints added)

3 Conceptual Modeling of Temporal Facts

At the conceptual level, and ORM in particular, basic *temporal object types* (e.g. Date or Period) may be used in models like other object types, with relevant temporal operators (e.g. $-$, overlaps) predefined for the type. For ORM, we introduce some useful classifications. A temporal object type is once-only or repeatable. A *once-only temporal object* is a single instant or period. Once-only types (e.g. Year(CE), Month(yr)) are useful for recording when an individual event (non-repeatable) happened or will happen (e.g. the election of the next US president). A *repeatable temporal object* corresponds to a set of instants/periods. Repeatable object types (e.g. Weekday(.code), MonthOfYear(.nr)) are useful for modeling schedules (e.g. a workout routine).

Periods may be modeled by explicitly indicating their start and end times (if known), using an external uniqueness constraint to provide an identifier. Durations may be modeled by a simple object type with a unit for the chosen temporal granularity, e.g. Age(y:).

For modeling purposes, a fact is a proposition taken to be true by the business, and a fact type is a set of possible fact instances. We classify fact types as definitional, once-only, or repeatable. *Definitional facts* are true by definition, so have no temporal aspect. For example, the fact type PolygonShape has NrSides is definitional. Each *once-only fact* corresponds to a single event. Its truth is determined by an event that can never be repeated in the business domain (e.g. Terry Halpin was born in Australia). So ignoring reincarnation, the fact type Person was born in Country is once-only. *Each repeatable fact* corresponds to a set of events. Its truth is determined by any one of a non-unit-set of repeatable events (e.g. Terry Halpin visited Mexico). So the fact type Person visited Country is repeatable. For each once-only or repeatable fact type in a model, we need to determine what (if any) temporal information is needed.

An event may be a *point event* (occurs at an instant) or a *period event* (has nonzero duration, e.g. your reading of this paper). For once-only fact types relating to point

events, if we wish to record when at least some instances of those events occurred, add a temporal fact type of the desired granularity (e.g., for Person was born in Country, add Person was born on Date, or Person was born in Year etc.). For once-only fact types relating to period events, to record when at least some instances of those events occurred, add temporal fact types of the desired granularity to note the start and end (if known) of the period (e.g. FirstReading started at Time(dhm), FirstReading ended at Time(dhm)). Here FirstReading may be modeled as an objectification of Person first read Paper, or as a coreferenced type identified by FirstReading is by Person, FirstReading is of Paper. If Period is explicitly introduced (e.g. FirstReading occupied Period) then the start and end predicates are attached to Period. If we are not interested in distinguishing start and end, we may model it as for a point event using coarse granularity (e.g. FirstReading occurred on Date).

While once-only fact types are unchangeable, repeatable fact types may be *changeable* (e.g. Patient has Temperature, Patient is allergic to Drug). For such fact types, if we are interested only in the current *snapshot* then no remodeling is needed (simply update the fact populations as required). To maintain *history* of a changeable fact type that is *functional*, we may simply insert into its key the relevant role played by a temporal object type of the desired granularity (e.g. Patient(nr) at Hour(dh) had Temperature(°C:)). This flattened approach may be remodeled using nesting or coreferencing in the usual way. For example, use the fact type TemperatureMeasurement recorded Temperature, where TemperatureMeasurement is either an objectification of Patient had temperature taken at Hour or is coreferenced by TemperatureMeasurement is of Patient and TemperatureMeasurement is at Hour. As a further alternative, a simple identifier may be introduced for the measurement object type, e.g. TemperatureMeasurement(nr).

To maintain *history of nonfunctional fact types* that are changeable, the previous patterns may be modified to include a distinguishing temporal role (e.g. startdate or starttime), to distinguish different events that make the same fact true. Consider for example, the report of country visits shown in Fig. 3. For each visit, the start date is known and possibly the end date is known (“?” denotes a null). Employee 102 visited The Netherlands twice, and we wish to retain a record of both visits, so we cannot model visits by the simple fact type Employee visited Country.

<i>Visit:</i>	<i>empNr</i>	<i>countryCode</i>	<i>startdate</i>	<i>enddate</i>
	101	NL	2000-01-01	2000-01-15
	101	CA	2008-02-15	?
	102	NL	2007-06-05	2007-06-20
	102	BE	2007-06-20	2007-06-25
	102	NL	2008-06-08	?

Fig. 3. Record of visits to countries by employees

Let us assume that for any given date, an employee may start visiting or end visiting at most one country (if this is not true, replace Date by Instant). Fig. 4 shows basic ORM schemas for this situation, in (a) nested, (b) coreferenced, and (c) flattened form. Other solutions are to introduce a simple identifier for Visit, or an ordinal number as part of the identifier (e.g. Fred’s 2nd visit to France). ORM’s current relational mapping algorithm (Rmap) maps (a) and (b) to (d), and (c) to (e).

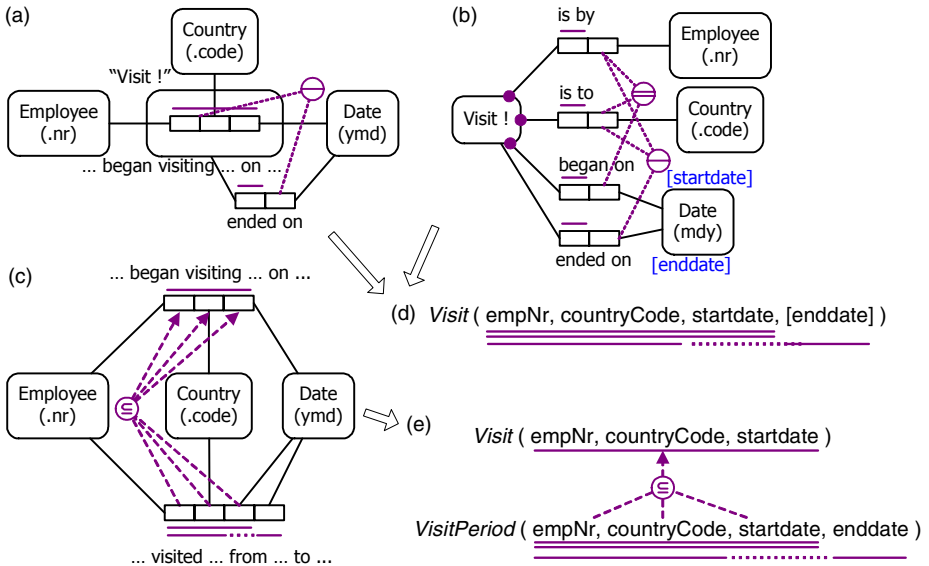


Fig. 4. Basic ORM schemas for modeling the data in Fig. 3

In verbalizing the data in the Fig. 3 report, it seems most natural to use a quaternary for row 1 and a ternary from row 2, leading to the flattened solution (Fig. 4(c)). But Rmap punishes the modeler for this choice by mapping to the 2-table relational schema (Fig. 4(e)) instead of the simpler 1-table schema (Fig. 4(d)) obtained from the nested or coreferenced schemas. As an enhancement to the NORMA tool [6] for ORM 2, we are modifying Rmap to allow retention of the flattened schema while still offering the single table relational map by default (the user may override this choice).

If we also want to talk about visits (e.g. to record the main purpose of a visit as business or pleasure), the nested or coreferenced solutions are far preferable (compare adding the fact type Visit is for VisitPurpose with adding the quaternary Employee began visiting Country on Date for main- VisitPurpose together with another 3-part subset constraint).

Two temporal constraints need to be added to the schemas in Fig. 4. The first is a value-comparison constraint that is most easily understood using schema Fig. 4(b). This constraint may be depicted graphically in ORM 2 [15, p. 290] or verbalized textually as: **For each** Visit, **existing** enddate ≥ startdate. The “existing” qualification applies the condition only where an end date does exist.

The second constraint requires that no two visits by the same employee overlap *nontrivially* in time (the data in rows 3 and 4 of Fig. 3 indicate that trivial overlap is allowed in this business domain). This constraint cannot be captured graphically in ORM 2 but can be specified textually in either static form or dynamic form. The static form is complex (cf. the restaurant seating example in [3]), whereas given the value-comparison constraint, the dynamic form of the overlap constraint may be rendered simply: **For each** Employee, **existing previous** Visit.enddate ≤ **added** Visit.startdate. For discussion on the semantics underlying such syntax, see [3].

Note that if we have *complete* knowledge of all visit periods by an employee, we could derive the quaternary in Fig. 4(c) from the two ternaries Employee began visiting Country on Date and Employee ended a visit to Country on Date, with a pair subset constraint between the Employee-Country role pairs (from the enddate fact type to the startdate fact type), by ordering visit periods sequentially. However, if we have incomplete knowledge we cannot derive the quaternary, and the two ternaries solution must be rejected (e.g. the ternary solution allows a population of the two tuples <101, NL, 2000-01-01, ?> and <101, NL, ?, 20008-02-15>, but the employee might have made two visits, not one visit. This raises a fine point about the notion of *elementarity* of facts. Assuming complete knowledge, and the derivation possibility by ordering visit periods, is the quaternary fact “Employee 101 visited the Country ‘NL’ from the Date ‘2000-01-01’ to the Date ‘2000-01-15’” elementary? We leave further investigation of this issue as a research topic.

Sometimes, business rules require no overlap (trivial or nontrivial). For example, in modeling pay awards, it is normal to require for each JobPosition that no two (start-date, enddate) periods overlap. And if we modify the country visit example to country habitation, where on a given date a person may start or end residing in at most one country, the country role is excluded from the identification scheme for habitations, and no overlap is allowed.

As a final note before ending this section, one difference between the ORM and CogNIAM (www.pna-group.com) flavors of fact-oriented modeling is that ORM forbids the inclusion of nulls in asserted (non-derived) facts. For example, ORM ignores the null in verbalizing the ternary fact on row 2 of Fig. 3, whereas CogNIAM allows this row to be verbalized as a quaternary including the null. In ORM we have found it useful to be able to specify additional constraints on derived fact types, where nulls are allowed in fact populations, but have found it safer to avoid nulls in asserted fact types (requiring any asserted fact to be either elementary or existential). Which of these approaches is better in this regard is left as a topic for further discussion.

4 Modeling History of Migration between Role Subtypes

In previous work [14], we outlined a general approach for modeling histories of entities as they migrated from one role subtype to another. In this section, after a brief review of some basic concepts, we now extend that work.

A type is *rigid* if each instance of it must remain in that type for the duration of that instances’s lifetime (e.g. Person, Tree), otherwise the type is a *role type* (e.g. Employee, Cricketer). Over time, an entity may move from one role type to another. Suppose each role has specific details of interest and we want to maintain this history of an entity as it changes roles. We now classify role subtypes as once-only or repeatable. With a *once-only role subtype*, objects can never return to play that role again once they have left the subtype (e.g. Child, SinglePerson). With a *repeatable role subtype* objects can return to play that role again (e.g. Employee, MarriedPerson).

Histories involving transitions between once-only role subtypes may be modeled using a *successive disjunctions pattern*. For example, Adult is a subtype of TeenagerOrAdult which in turn is a subtype of ChildOrTeenagerOrAdult. Subtype specific details may now be easily retained (e.g. Adult has favorite- Book, TeenagerOrAdult as a teen

had favorite- PopGroup, ChildOrTeenagerOrAdult as a child had favorite- Toy). This arrangement automatically caters for the linear transition order from role to role.

If the roles are once-only, then an alternative solution is to use what we call the *once-only role playing pattern*, augmented by a *dynamic constraint* to constrain the possible role transitions. For example, the child-teenager-adult example may be modeled as shown in Fig. 5.

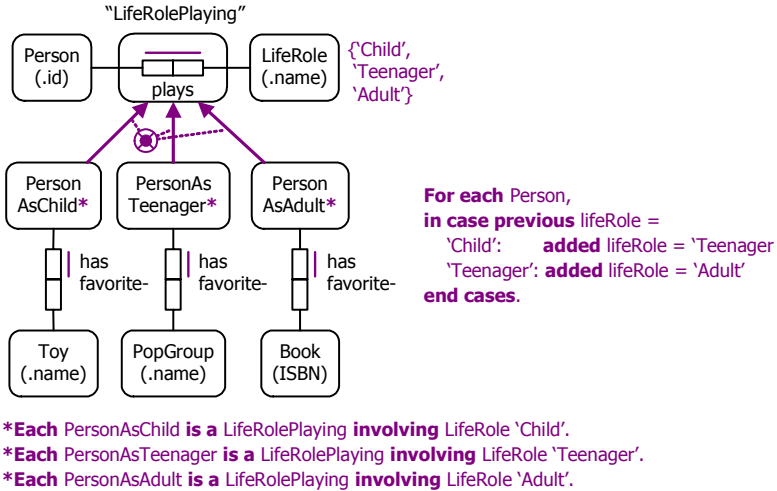


Fig. 5. Example of once-only role playing pattern with dynamic rule

If a role subtype is repeatable, the previous approaches cannot record history of multiple playings of the same role by the same object. To address this problem, we provide what we call the *repeatable role playing pattern*, which includes the start-time of a role playing as part of its natural identifier. One version of this is shown in Fig. 6 (minus the dynamic rule). This assumes that a person may begin or end a given role at most once on the same date (if this is not true, replace Date by Instant). This pattern allows that a person may begin or end multiple roles on the same date. Alternative versions of the pattern introduce either simple identifiers, or ordinal numbers as partial identifiers, for RolePlaying. A concrete example is given in Fig. 7.

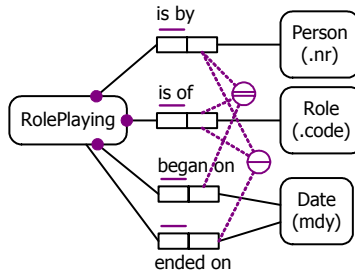


Fig. 6. One version of the repeatable role playing pattern

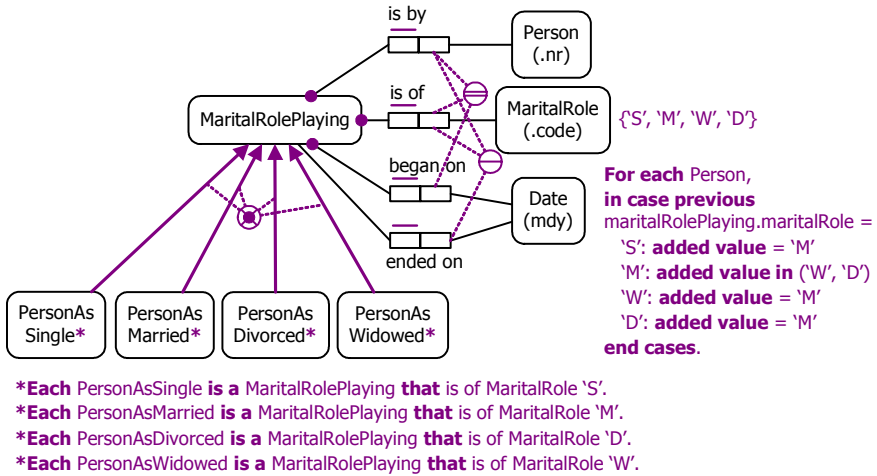


Fig. 7. Example of repeatable role playing pattern with dynamic rule

5 Conclusion

This paper reviewed some approaches to temporal data and operations, identified problems with Allen's operators and OWL-Time, suggested conceptual ways to classify temporal information, raised some issues regarding elementarity, and proposed modeling heuristics and data model patterns to address time-impacted tasks such as modeling histories, and tracking entities across time as they migrate between roles.

While the graphic depiction of ORM models has been implemented in the NORMA tool, the detailed syntax for textual specification of temporal and dynamic rules (including scheduling) and the generation of code from such textual rules is still a work in progress. We plan to extend the NORMA tool to support such rules, and also implement a mapping from ORM to OWL, work on which has already begun. It may also be worthwhile considering graphical extensions to ORM to directly support some temporal aspects (e.g. marking types as once-only or repeatable).

References

1. Allen, J.: Maintaining Knowledge about Temporal Intervals. *Communications of the ACM* 26(11), 832–843 (1983)
2. Bakema, G., Zwart, J., van der Lek, H.: *Fully Communication Oriented Information Modelling*, Ten Hagen Stam, The Netherlands (2000)
3. Balsters, H., Carver, A., Halpin, T., Morgan, T.: Modeling Dynamic Rules in ORM. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops*. LNCS, vol. 4278, pp. 1201–1210. Springer, Heidelberg (2006)
4. Bruza, P.D., van der Weide, T.P.: *The Semantics of TRIDL*, Technical Report 89-17, Department of Information Systems, University of Nijmegen (1989)
5. Chen, P.P.: The entity-relationship model—towards a unified view of data. *ACM Transactions on Database Systems* 1(1), 9–36 (1976)

6. Curland, M., Halpin, T.: Model Driven Development with NORMA. In: Proc. 40th Int. Conf. on System Sciences (HICSS-40), IEEE Computer Society, Los Alamitos (2007)
7. Date, C., Darwen, H., Lorentzos, N.: Temporal Data and the Relational Model. Morgan Kaufmann, San Francisco (2003)
8. Edelweiss, N., de Oliveira, J., de Castilho, J., Montanari, E., Pernici, B.: T-ORM: Temporal aspects in objects and roles. In: Halpin, T., Meersman, R. (eds.) Proc. First International Conference. on Object-Role Modeling, University of Queensland, pp. 18–27 (1994)
9. Embley, D.W.: Object Database Development. Addison-Wesley, Reading (1998)
10. Falkenberg, E.D., van der Weide, T.P.: Formal Description of the TOP Model. Technical Report 88-01, Department of Information Systems, University of Nijmegen (1988)
11. Halpin, T.: Information Modeling in UML and ORM: A Comparison. In: Khosrow-Pour, M., Hershey, I.G.I. (eds.) Enc. of Inf. n Science and Technology, vol. 3, pp. 1471–1475 (2005)
12. Halpin, T.: ORM 2. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2005, vol. 3762, pp. 676–687. Springer, Heidelberg (2005)
13. Halpin, T.: ORM/NIAM Object-Role Modeling. In: Bernus, P., Mertins, K., Schmidt, G. (eds.) Handbook on Information Systems Architectures, 2nd edn., pp. 81–103. Springer, Heidelberg (2006)
14. Halpin, T.: Subtyping Revisited. In: Pernici, B., Gulla, J. (eds.) Proc. CAiSE 2007 Workshops, vol. 1, pp. 131–141. Tapir Academic Press, London (2007)
15. Halpin, T., Morgan, T.: Information Modeling and Relational Databases, 2nd edn. Morgan Kaufmann, San Francisco (2008)
16. Halpin, T., Wagner, G.: Modeling Reactive Behavior in ORM. In: Conceptual Modeling – ER 2003, Proc. 22nd ER Conference, Chicago. LNCS, Springer, Heidelberg (2003)
17. ter Hofstede, A.H.M.: Information Modelling in Data Intensive Domains, PhD thesis, University of Nijmegen (1993)
18. ter Hofstede, A.H.M., Proper, H.A., van der Weide, T.P.: Formal definition of a conceptual language for the description and manipulation of information models. Information Systems 18(7), 489–523 (1993)
19. ISO 2004, ISO 8601:2004(E): Data elements and interchange formats—Information interchange—Representation of dates and times. ISO, Geneva (2004)
20. Object Management Group (2003), UML 2.0 Superstructure Specification, <http://www.omg.org/uml>
21. Object Management Group (2005), UML OCL 2.0 Specification, <http://www.omg.org/docs/ptc/05-06-06.pdf>
22. Parent, C., Spaccapietra, S., Zimanyi, E.: Conceptual Modeling for Traditional and Spatio-Temporal Applications. Springer, Berlin (2006)
23. Petrounias, I., Loucopoulos, P.: Time Dimension in a Fact-Based Model. In: Halpin, T., Meersman, R. (eds.) Proc. First International Conference on Object-Role Modeling, Key Centre for Software Technology, University of Queensland, pp. 1–17 (1994)
24. Proper, H.A.: A Theory for Conceptual Modeling of Evolving Application Domains, PhD thesis, University of Nijmegen (1994)
25. Snodgrass, R.: Developing Time-Oriented Database Applications in SQL. Morgan Kaufmann, San Francisco (2000)
26. Sowa, J.: Knowledge Representation. Brooks/Cole, Pacific Grove (2000)
27. Theodoulidis, C., Loucopoulos, P., Kopanas, V.: A Rule Oriented Formalism for Active Temporal Databases. In: Lyytinen, K., Tahvanainen, V.-P. (eds.) Next Generation CASE Tools, IOS Press, Amsterdam (1992)
28. Theodoulidis, C., Wangler, B., Loucopoulos, P.: The Entity-Relationship-Time Model. In: Conceptual Modelling, Databases, and CASE: An Integrated View of Information Systems Development, ch. 4, pp. 87–115. John Wiley & Sons, Chichester (1992)
29. Wintraecken, J.: The NIAM Information Analysis Method: Theory and Practice. Kluwer, Deventer (1990)

Formal Semantics of Dynamic Rules in ORM

Herman Balsters¹ and Terry Halpin²

¹ University of Groningen, The Netherlands
H.Balsters@rug.nl

² Neumont University, Utah, USA
terry@neumont.edu

Abstract. This paper provides formal semantics for an extension of the Object-Role Modeling approach that supports declaration of dynamic rules. Dynamic rules differ from static rules by pertaining to properties of state transitions, rather than to the states themselves. In this paper we restrict application of dynamic rules to so-called single-step transactions, with an old state (the input of the transaction) and a new state (the direct result of that transaction). These dynamic rules further specify an elementary transaction type by indicating which kind of object or fact (being added, deleted or updated) is actually allowed. Dynamic rules may declare pre-conditions relevant to the transaction, and a condition stating the properties of the new state, including the relation between the new state and the old state. In this paper we provide such dynamic rules with a formal semantics based on sorted, first-order predicate logic. The key idea to our solution is the formalization of dynamic constraints as static constraints on the database transaction history.

1 Introduction

Object-Role Modeling (ORM) is a fact-oriented approach for modeling, transforming, and querying information in terms of the underlying facts of interest, where facts and rules are verbalized in language understandable by nontechnical users of the business domain. In contrast to attribute-based modeling approaches such as Entity Relationship (ER) modeling [5] and class diagramming in the Unified Modeling Language (UML) [18], ORM models are attribute-free, treating all facts as relationships (unary, binary, ternary etc.). For example, instead of the attributes `Person.isSmoker` and `Person.birthdate`, ORM uses the fact types `Person smokes` and `Person was born on Date`.

Other fact-oriented approaches closely related to ORM include CogNIAM (www.pna-group.com), Fully-Communication Oriented Information Modeling (FCO-IM) [2], and the Semantics of Business Vocabulary and Business Rules (SBVR) [20] specification recently approved by the Object Management Group. A basic introduction to ORM may be found in [12] and a thorough coverage in [13]. The version of ORM discussed in this paper is ORM 2 [11], as supported by the NORMA tool [7].

Business rules include constraints and derivation rules. *Static rules* (also known as state rules) apply to each state of the information system that models the business domain, and may be checked by examining each state individually (e.g. each moon

orbits at most one planet). *Dynamic rules* reference at least two states, which may be either successive (e.g. no employee may be demoted in rank—this kind of dynamic rule is known as a transition constraint) or separated by some period (e.g. invoices ought to be paid within 30 days of being issued). ORM is richer than ER or UML in its ability to depict static constraints graphically, but unlike UML it currently has no graphic notation (e.g. activity diagrams) to specify business processes. To capture dynamic rules, UML supplements its graphical notations with formulae in the Object Constraint Language (OCL) [19, 25], but the OCL syntax is often too mathematical for validation by nontechnical users.

Since the 1980s, many extensions to fact-oriented approaches have been proposed to model temporal aspects and processes (e.g. [4, 8, 14, 15, 16, 22, 23]). For a brief review of such work see [1], where we in conjunction with two colleagues introduced to ORM a purely declarative means to formulate dynamic constraints on *single-step transactions*, with an old state (the input of the transaction) and a new state (resulting from that transaction). Such dynamic rules specify an *elementary transaction type* indicating which kind of object or fact is being added, deleted or updated, and (optionally) pre-conditions relevant to the transaction, followed by a condition stating the properties of the new state, including the relation between the new state and the old state. These dynamic rules are formulated in a syntax designed to be easily validated by nontechnical domain experts. In this paper, we focus on providing a formal semantics for the basic rule patterns for dynamic rules found in [1]. Such a formalization supports further understanding of dynamic rules, and also provides a step to further tool support.

Substantial research has been carried out to provide logical formalizations of dynamic rules, typically using temporal logics (e.g. [9], ch. 8) or Event-Condition-Action (ECA) formalisms (e.g. de Brock [3], Lipeck [17], Chomicki [6], Paton & Díaz [21], Snodgrass[24]). Our approach differs from previous work by *treating a dynamic rule as a special kind of static rule on the transaction history*. We define the semantics of a dynamic rule by making explicit the log of all previous transaction instances pertaining to that specific rule. Snapshot data are maintained in the user database, whereas historical data are kept in the log database. This *logging semantics* allows us to formalize dynamic constraints in a static way, using first-order predicate logic. ORM model fragments associated with dynamic rules may also be fully formalized in this way, as first described in [10] using unsorted predicate logic; for ease of readability, we now use sorted predicate logic. This enables us to offer the full semantics of ORM models, including both static and dynamic rules, in one coherent framework.

The rest of this paper is structured as follows. Section 2 provides a simple example of how a graphical ORM model (with no dynamic rules) may be transformed into a logical theory. Section 3 shows how to formalize dynamic rules over updates to single-valued roles in functional fact types. Section 4 extends this case to capture history. Section 5 considers additions of instances to nonfunctional fact types. Section 6 examines a more complex case involving derivation. Section 7 summarizes the main contributions, notes some further research options, and provides a list of cited references for further reading.

2 Formalizing Basic ORM Models as Logical Theories

An ORM model includes both schema (structure) and population (instances). In [10], one of us provided a detailed algorithm for translating any ORM model into a set of formulae in unsorted predicate logic (with identity, and using mixfix predicates and numeric quantifiers). We now use basically the same approach, but employ sorted logic. While there is no space here to cover the full algorithm, we illustrate the basic approach with a simple example.

The ORM model shown in Fig. 1 includes a schema with one elementary fact type *Employee has Salary* and a population of three fact instances. Fig. 1(a) is in compact form, abbreviating the reference schemes for *Employee* and *Salary* in parentheses. These reference schemes may be automatically expanded to the existential fact types shown in Fig. 1(b). In ORM 2, unit-based reference schemes (e.g. USD:) also involve a unit dimension (in this case, Money) but for simplicity we ignore this aspect here.

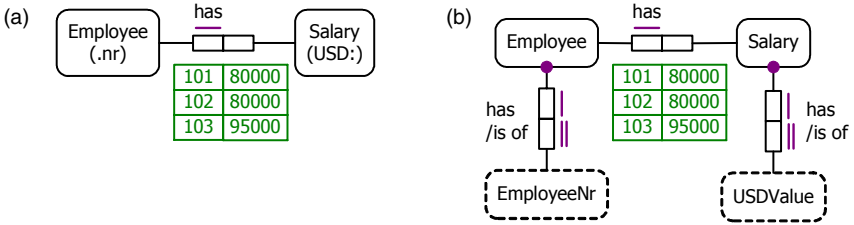


Fig. 1. A simple ORM model in (a) compact and (b) expanded form

The model may be formalized as indicated below. Object types are typed as entity types (solid line) or value types (broken line), and the top level entity types are declared mutually exclusive. For simplicity, we omit classifications of value types here and relevant axioms for numeric operators. The predicates are then typed. Although our sorted logic notation uses short predicate names (e.g. “has”), different fact types are always distinguished by typing the object variables. If one wishes to avoid predicates with different semantics being assigned the same short name (cf. Horse runs Race with Person runs Company), full fact type readings may be used instead to name the predicates. Type predicates are placed in prefix position; all other predicates are mixfix.

Object Types: $\forall x:\text{Employee } x \text{ is an entity; } \forall x:\text{Salary } x \text{ is an entity}$
 $\forall x:\text{EmployeeNr } x \text{ is a value; } \forall x:\text{USDValue } x \text{ is a value}$
 $\forall x:\text{Employee } \forall y:\text{Salary } x \neq y$

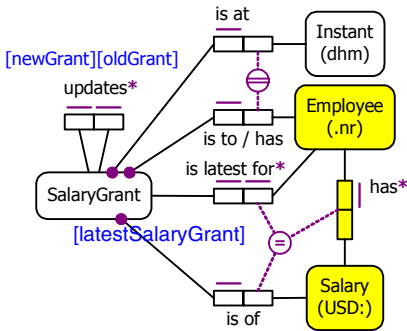
Fact Types: $\forall x \forall y (x \text{ has } y \rightarrow [(\text{Employee } x \ \& \ \text{Salary } y) \vee (\text{Employee } x \ \& \ \text{EmployeeNr } y)$
 $\vee (\text{Salary } x \ \& \ \text{USDValue } y)]$
 $\forall x:\text{Employee } \forall y:\text{EmployeeNr } (x \text{ has } y \equiv y \text{ is of } x)$
 $\forall x:\text{Salary } \forall y:\text{USDValue } (x \text{ has } y \equiv y \text{ is of } x)$

Constraints: $\forall x:\text{Employee } \exists^{0..1}y:\text{Salary } x \text{ has } y$
 $\forall x:\text{Employee } \exists^1y:\text{EmployeeNr } x \text{ has } y$
 $\forall x:\text{EmployeeNr } \exists^{0..1}y:\text{Employee } x \text{ is of } y$
 $\forall x:\text{Salary } \exists^1y:\text{USDValue } x \text{ has } y; \forall x:\text{USDValue } \exists^{0..1}y:\text{Salary } x \text{ is of } y$

Population: $\exists x:\text{Employee} \exists y:\text{Salary} \exists z:\text{EmployeeNr} \exists w:\text{USDValue}$
 (x has y & x has z & y has w & z = 101 & w = 80000)
 etc. for the other 2 rows of data

3 Updating Single-Valued Roles in a Functional Fact Types

We now consider formalization of dynamic rules added to ORM models, starting with the case of updates to a functional ($n:1$ or $1:1$) binary fact type. The dynamic constraint on the salary fact type in Fig. 1 requires that salaries of employees must not decrease. Using the syntax introduced in [1], where **old** and **new** to refer to situations immediately before and after the transition, this constraint may be stated textually as: **For each Employee, new salary \geq old salary.** Here the *context* of the constraint is the object type Employee, and the elementary transaction *updates* the salary of the employee.



Derivation rule for the snapshot fact type is provided by the equality constraint which may be formulated in attribute form thus:

* **For each Employee,**
 salary = latestSalaryGrant.salary.

* SalaryGrant is latest for Employee **iff**
 SalaryGrant is to Employee **and** is at **some** Instant₁
and not exists some SalaryGrant₂ **that** is to the **same** Employee **and** is at **some** Instant₂ > Instant₁.

* SalaryGrant₂ updates SalaryGrant₁ **iff**
 SalaryGrant₁ is to Employee₁ **and** SalaryGrant₂ is to Employee₁
and SalaryGrant₁ is at Instant₁ **and** SalaryGrant₂ is at Instant₂
and not exists some SalaryGrant₃ **that** is to Employee₁ **and** is at Instant₃
and Instant₃ > Instant₁ **and** Instant₃ < Instant₂.

Fig. 2. Logging semantics for update salary rule

While the user schema is confined to Employee has Salary, for which only a current snapshot is required (no history), in the background we add fact types to maintain a log of salary grants, as shown in Fig. 2 (unshaded portion). The strict order on Instant enables us to define the notions of *latest* as well as *updating* of an old salary grant by a new one as shown. The employee-salary fact type is now derivable from the equality constraint, as shown. The dynamic constraint may now be reformulated as the following static constraint (no action is needed if the salary grant is the first for the employee): SalaryGrant₂ updates SalaryGrant₁ **only if** SalaryGrant₂.salary \geq SalaryGrant₁.salary.

The additional object types and fact types may be formalized as discussed in the previous section. The graphical constraints are also trivially formalized. For example, the external uniqueness constraint and the join equality constraint are expressible as:

$$\forall x:\text{Instant} \forall y:\text{Employee} \exists^0..1 z:\text{SalaryGrant} (z \text{ is at } x \ \& \ z \text{ is to } y)$$

$$\forall x:\text{Employee} \forall y:\text{Salary} [x \text{ has } y \equiv \exists z:\text{SalaryGrant}(z \text{ is latest for } x \ \& \ z \text{ is of } y)]$$

The derivation rules in Fig. 2 are expressed in FORML 2, our formal ORM 2 textual language that is a sugared version of our underlying logical syntax designed for consumption by nontechnical domain experts. Type names are used for sorted variables, with subscripts added as needed to distinguish variables of the same type. Where not stated explicitly, head clause variables are implicitly universally quantified, and variables introduced in the body clause are existentially quantified (cf. Horn clauses). Functional style in dot notation may be used, using role names as function names. For example, the join equality constraint formulated above may be reformulated in functional style as

$$\forall x:\text{Employee} \ x.\text{salary} = x.\text{latestSalaryGrant}.\text{salary}$$

and then sugared to the FORML 2 rule: **For each** Employee, salary = latestSalaryGrant.salary. The other derivation rules in Fig. 2 are equivalent to the following:

$$\forall x:\text{SalaryGrant} \forall y:\text{Employee} [x \text{ is latest for } y \equiv (x \text{ is to } y \ \& \ \exists z:\text{Instant} \ x \text{ is at } z \ \& \ \sim \exists w:\text{SalaryGrant} \ \exists u:\text{Instant} \ (w \text{ is to } y \ \& \ u > z))]$$

$$\forall x,y:\text{SalaryGrant} [y \text{ updates } x \equiv (x.\text{employee} = y.\text{employee} \ \& \ \sim \exists z:\text{SalaryGrant} (z.\text{employee} = x.\text{employee} \ \& \ y.\text{instant} > z.\text{instant} \ \& \ z.\text{instant} > x.\text{instant}))]$$

The key result is that the dynamic constraint **For each** Employee, **new** salary \geq **old** salary may be recast as the following static constraint:

$$\forall x,y:\text{SalaryGrant} (y \text{ updates } x \rightarrow y.\text{salary} \geq x.\text{salary})$$

Generalizing from this example to any functional binary fact type of the form $A R's B$, with B 's role name r (denoting the “attribute” of A being constrained), we obtain the dynamic constraint pattern **For each** A , **new** $r \Theta$ **old** r , where Θ denotes the required relationship between the values of r after and before the transition. Our logic specifications for the salary example may be easily adapted to cover this general pattern.

This approach may be easily extended to formalize simple state transition rules such as the dynamic rule for marital status transitions shown in Fig. 3. As in the previous example, the presence of the **new** and/or **old** keywords signals that the prospective transaction is an update (rather than an addition or deletion). In this case, the logging subschema (unshaded portion) is based on MaritalStatusAssignment. The formalization is similar to that in the previous example, allowing the dynamic rule to be reformulated as the following static rule. This example may be generalized to updates of an enumerated role on a functional fact type $A R's B$ in an obvious way.

$$\forall x,y:\text{MaritalStatusAssignment} [y \text{ updates } x \rightarrow ((x.\text{maritalStatus} = \text{'single'} \ \& \ y.\text{maritalStatus} = \text{'married'}) \vee (x.\text{maritalStatus} = \text{'married'} \ \& \ (y.\text{maritalStatus} = \text{'widowed'} \ \vee \ y.\text{maritalStatus} = \text{'divorced'})) \vee (x.\text{maritalStatus} = \text{'widowed'} \ \& \ y.\text{maritalStatus} = \text{'married'}) \vee (x.\text{maritalStatus} = \text{'divorced'} \ \& \ y.\text{maritalStatus} = \text{'married'}))]$$

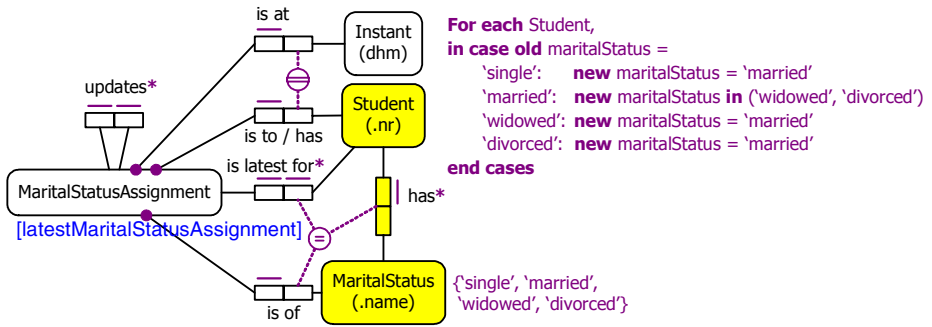


Fig. 3. Updating the marital status of students

4 Examples of Historical Facts

We now extend the salary snapshot case considered earlier to the case where salary *history* is required in the user schema. The dynamic constraint is now specified using the keywords “**added**”, “**previous**” and “**existing**”, as shown in Fig. 4(a) (for simplicity, reference schemes are omitted). The **added** keyword indicates we are adding a fact rather than updating an existing fact. The “**previous**” function returns the previous salary (if it exists) of the employee, while the qualification “**existing**” applies the condition only if a previous salary for the employee does exist.

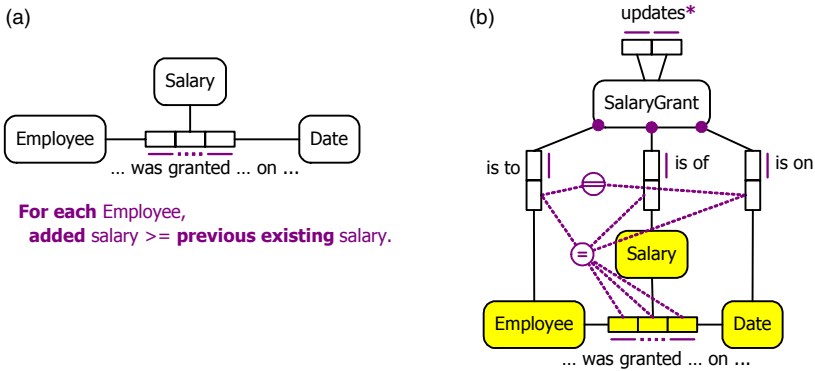


Fig. 4. Salary example with history

This dynamic rule syntax is much simpler than that used in [1], but still relies on a formal semantics being provided for the keywords. To address this need, we first objectify the ternary as SalaryGrant. In ORM 2, this is handled as situational nominalization [13, sec. 10.5], with SalaryGrant in 1:1 correspondence with the ternary, as enforced by the join equality constraint in Fig. 4(b). The case may now be handled as in the earlier case, with Date replacing Instant. In effect, this case is simpler, because the logging data is already available from the application database.

This example may be generalized to any historical fact type of the form $R(A_1.. A_n)$, where a uniqueness constraint spans $n-1$ roles, one of which is played by a temporal object type such as Date or Instant that is used to order the history.

5 Adding Instances of a Nonfunctional Fact Type

We now consider *adding fact instances* to a *nonfunctional fact type* (no single-role uniqueness constraint), such as the Seating was allocated Table association in Fig. 5, which shows a model fragment extracted from a restaurant application. A seating is the allocation of a party (of one or more customers) to one or more vacant tables. The asterisked rule is a derivation rule for the snapshot fact type Table is vacant.

This model maintains a *history* of seatings (for each table we record all the seatings it was previously allocated to). The value-comparison constraint (circled “>” with dots) verbalizes as: **For each** Seating, **existing** endTime > startTime. To ensure that no seatings that overlap in time occupy the same table, the dynamic rule in Fig. 5 declares that a table may be assigned to a seating only if it is vacant *at that time*. The *context* for the constraint is the fact type Seating was allocated Table, and the elementary transaction involves the *addition* of an *instance* of this fact type. The reserved words **before** and **after** denote the states just before and after the transaction, **needed** indicates the precondition is necessary for the fact addition to take place (not just for this constraint), and **the** is scoped to the transaction instance.

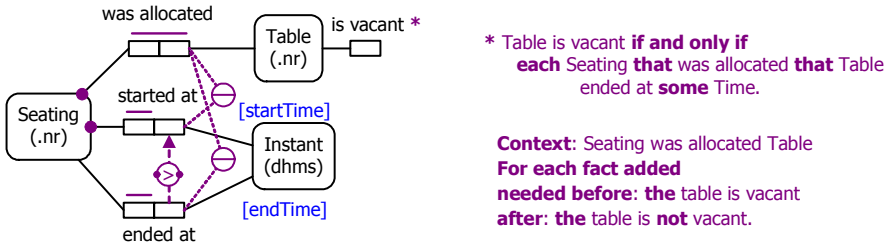


Fig. 5. Fragment of an ORM schema maintaining history of restaurant seatings

One static alternative to the dynamic rule was presented in [1], but this expression is extremely complex. A simpler static constraint formulation is possible using our logging semantics approach. The unshaded portion of Fig. 6 introduces TableSeating in 1:1 correspondence with Seating was allocated to Table, using the derivation rules shown to determine its start and end times. The value comparison constraint **For each** TableSeating, **existing** endtime > starttime is omitted since it is implied. The updates and latest table seating predicates may be defined similarly to the previous examples, where one table seating updates another if and only if both seatings are for the same table and there is no intermediate seating for that table. Given the value comparison constraint, the dynamic rule to ensure no overlap may now be simply formulated as the following static constraint:

$$\forall x,y:\text{TableSeating} (y \text{ updates } x \rightarrow y.\text{startTime} > x.\text{endTime})$$

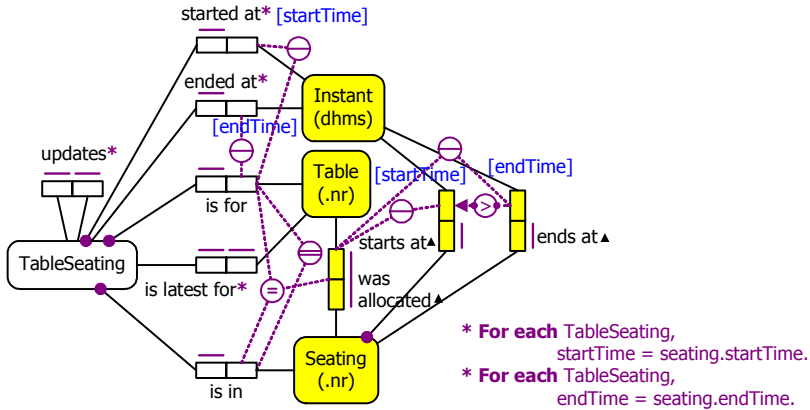


Fig. 6. Adding the logging subschema for Fig. 5

6 A More Complex Case Involving Derivation

In [1], a more complex case dealt with account transactions. We have space here to consider only transfer transactions, a basic schema for which is shown in Fig. 7. Transfer transactions transfer funds from one account to another. We record historical information of all transactions, from which the current account balances may be derived. We assume that an account exists prior to any transaction on it, and that on the event that an account is opened, its balance is set to zero. The following dynamic constraint may be specified on transfer transactions:

Context: TransferTransaction

For each instance added

newFromBalance = (old fromAccount.balance - amount) and

newToBalance = (old toAccount.balance + amount) and

new fromAccount.balance = newFromBalance and

new toAccount.balance = newToBalance

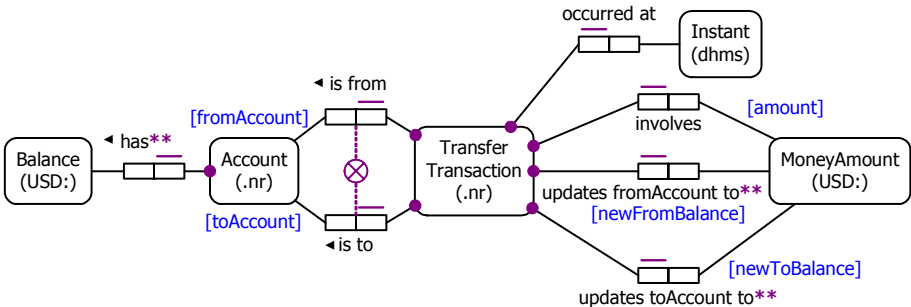


Fig. 7. An example involving historical and derived snapshot data

As with normal banking operations, history is maintained for all transactions in the user database, so the data needed for our logging semantics are essentially already there. Similar techniques to those discussed earlier can be applied to replace the dynamic rule with a static constraint.

7 Conclusion

This paper outlined an approach to provide a formal semantics for a proposed extension to Object-Role Modeling that supports declaration of dynamic rules. Dynamic rules differ from static rules by pertaining to properties of state transitions, rather than to the states themselves. We have restricted application of dynamic rules to so-called single-step transactions. Dynamic rules further specify an elementary transaction type by indicating which kind of object or fact (being added, deleted or updated) is actually allowed. Dynamic rules are equipped with preconditions relevant to the transaction, followed by a condition stating the properties of the new state, including the relation between the new state and the old state.

We have provided such dynamic rules with a formal semantics based on sorted, first-order predicate logic. The key idea to our solution is the formalization of dynamic constraints as static constraints on the database transaction history. This *logging semantics* for dynamic rules makes explicit the log of all previous transaction instances pertaining to those specific rules. Snapshot data are maintained in the user database, whereas historical data are kept in the log database. This approach avoids the need to consider more complex logics such as temporal logics, while at the same time conforming in part to industrial database approaches that utilize log files to manage transactions. Future research options include extending this framework to cover other kinds of transactions (e.g. deletions) as well as dynamic rules involving more complex temporal expressions.

References

1. Balsters, H., Carver, A., Halpin, T., Morgan, T.: Modeling Dynamic Rules in ORM. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1201–1210. Springer, Heidelberg (2006)
2. Bakema, G., Zwart, J., van der Lek, H.: Fully Communication Oriented Information Modelling, Ten Hagen Stam, The Netherlands (2000)
3. de Brock, E.O.: A General Treatment of Dynamic Integrity Constraints. *Data and Knowledge Engineering* 32(3), 223–246 (2000)
4. Bruza, P.D., van der Weide, T.P.: The Semantics of TRIDL, Technical Report 89-17, Department of Information Systems, University of Nijmegen (1989)
5. Chen, P.P.: The entity-relationship model—towards a unified view of data. *ACM Transactions on Database Systems* 1(1), 9–36 (1976)
6. Chomicki, J.: History-less Checking of Dynamic Integrity Constraints. In: ICDE 1992, pp. 557–564 (1992)
7. Curland, M., Halpin, T.: Model Driven Development with NORMA. In: Proc. 40th Int. Conf. on System Sciences (HICSS-40). IEEE Computer Society, Los Alamitos (2007)

8. Falkenberg, E.D., van der Weide, T.P.: Formal Description of the TOP Model. Technical Report 88-01, Department of Information Systems, University of Nijmegen (1988)
9. Girle, R.: Possible Worlds. McGill-Queen's University Press, Montreal (2003)
10. Halpin, T.: A Logical Analysis of Information Systems: static aspects of the data-oriented perspective, doctoral dissertation, University of Queensland (1989), http://www.orm.net/Halpin_PhDthesis.pdf
11. Halpin, T.: ORM 2, On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2005, vol. 3762, pp. 676–687. Springer, Heidelberg (2005)
12. Halpin, T.: ORM/NIAM Object-Role Modeling. In: Bernus, P., Mertins, K., Schmidt, G. (eds.) Handbook on Information Systems Architectures, 2nd edn., pp. 81–103. Springer, Heidelberg (2006)
13. Halpin, T., Morgan, T.: Information Modeling and Relational Databases, 2nd edn. Morgan Kaufmann, San Francisco (2008)
14. Halpin, T., Wagner, G.: Modeling Reactive Behavior in ORM. In: Song, I.-Y., Liddle, S.W., Ling, T.-W., Scheuermann, P. (eds.) ER 2003. LNCS, vol. 2813, pp. 567–569. Springer, Heidelberg (2003)
15. ter Hofstede, A.H.M.: Information Modelling in Data Intensive Domains, PhD thesis, University of Nijmegen (1993)
16. ter Hofstede, A.H.M., Proper, H.A., van der Weide, T.P.: Formal definition of a conceptual language for the description and manipulation of information models. Information Systems 18(7), 489–523 (1993)
17. Lipeck, U.W.: Transformation of Dynamic Integrity Constraints into Transaction Specifications, Theor. Comput. Sci. 76(1), 115–142 (1990)
18. Object Management Group 2003, UML 2.0 Superstructure Specification (2003), <http://www.omg.org/uml>
19. Object Management Group 2005, UML OCL 2.0 Specification (2005), <http://www.omg.org/docs/ptc/05-06-06.pdf>
20. Object Management Group 2007, Semantics of Business Vocabulary and Business Rules (SBVR) Specification (2007), http://omg.org/technology/documents/bms_spec_catalog.htm#SBVR
21. Paton, N.W., Díaz, O.: Active Database Systems. ACM Computing Surveys 31(1), 63–103 (1999)
22. Proper, H.A.: A Theory for Conceptual Modeling of Evolving Application Domains, PhD thesis, University of Nijmegen (1994)
23. Proper, H.A., Hoppenbrouwers, S.J.B.A., van der Weide, T.P.: A Fact-Oriented Approach to Activity Modeling. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2005. LNCS, vol. 3762, pp. 666–675. Springer, Heidelberg (2005)
24. Snodgrass, R.T.: TSQL2Language specification. SIGMOD Record 23(1), 65–86 (1994)
25. Warmer, J., Kleppe, A.: The Object Constraint Language, 2nd edn. Addison-Wesley, Reading (2003)

Fact Orientation and SBVR: The Catalyst for Efficient and Fully Integrated Education

Jos Vos

PBNA University and PNA University, The Netherlands
josjwvos@msn.com

Abstract. Fact orientation has a much broader use than solely IT. This paper describes the use of fact orientation and SBVR as the integration and educational method of a full bachelor degree program. The author describes advantages and results of the use of fact orientation in the educational area and recommends this approach for other subjects as well.

1 Introduction

There is a historical trend towards more integration. The industrial age was the generator of many stovepipes or silos as they are called in different speech communities. With the transition towards a knowledge economy one can observe a gradual trend towards more integration and fewer silos. More customer orientation by companies, more student orientation by educational institutes, more patient orientation by the most respected example of silos, the hospital, are clear examples of an increase in integration.

It is the expectation of the author that education is the next important candidate to offer more integration, resulting in accelerated and integrated education.

An important question is: can fact orientation and SBVR [9] help to achieve this? The author has been involved for about 18 years in the experiments in The Netherlands towards increased integration in bachelor education. In this paper the author describes the major components of an accelerated and fully integrated bachelor program, based on fact orientation, including concept definitions and at least three kinds of fact type forms. The program was officially accredited in 2006 by the Dutch-Flemish Accreditation Organisation (NVAO) [8].

We will, in the interest of brevity, abstract from the various intermediate historical results and discuss the contents of the accredited bachelor program and its foundation on fact orientation as defined above and SBVR.

More and more persons come to the conclusion that fact orientation can be used in teaching many other subjects [1, 7, 11]. More and more persons indicate that SBVR has a broader usage than IT [2, 3, 4, 6]. There are also persons that argue in favour of controlled natural language to replace the usual logic notation [10].

2 The Bachelor Program

The bachelor program is primarily intended for students that already have a job in an information or knowledge intensive organisation. This means that the students are awarded 80 (European) credit points of the required total of 240 for their practical experience. The remaining 160 credit points are obtained by taking 16 courses.

The first 4 courses are mandatory for each student, see figure 1. They are called Fact Orientation 101 (FO101), 102 (FO102), 103 (FO103) and 104 (FO104). The actual names are different in Dutch.

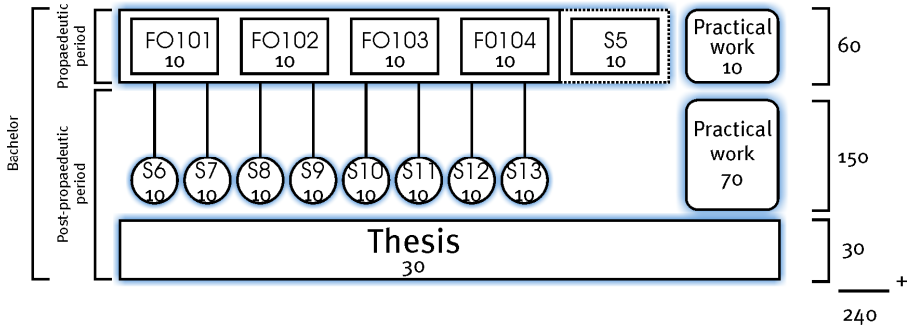


Fig. 1. The structure of the integrated bachelor

In Fact Orientation 101 the new book by Halpin and Morgan [5] (with the exception of the UML, SQL and process chapters) could be an excellent textbook. An example of a case used in the model exam is given below:

Case COMIT	
1	<i>To:</i> Jan de Visser
2	<i>From:</i> Frits de Hoog,
3	Manager HRM
4	<i>Subject:</i> Information need committees
5	
6	Dear Jan,
7	
8	To answer your question regarding the information needs in my department concerning committees and employees, I inform you about the following:
9	
10	
11	Each employee in our company and hence also in my department, is uniquely
12	identified by an employee number consisting of maximally 4 digits. This is the
13	only unique identification of an employee.
14	
15	Furthermore, we register for each employee the family name (consisting of maxi-
16	mally 20 characters), the first name (consisting of maximally 15 characters), the
17	birth date (in the notation mmddyyyy) and the gender (indicated by 'male' or
18	'female'). In addition to this, for every specific employee, the number of commit-
19	tee memberships is calculated, and the result is registered. It has to be noted that
20	being the chairman of a committee is not considered as a membership of that same
21	committee.
22	
23	Each committee is uniquely identified by its name, consisting of a maximum of 25
24	characters.
25	
26	Furthermore, for every committee the start date (in the notation mmddyyyy) and

27 the budget in Euros (consisting of maximally 7 digits) are registered, as well as the
 28 total number of members of the committee.
 29
 30 We only wish to mention a committee in our administration if certain precondi-
 31 tions are met:
 32 1. the chairperson is known,
 33 2. at least one member is known.
 34
 35 Each committee has only one chairperson.
 36 A committee can have many members.
 37 It is a rule in our company that an employee can only be the chairperson of one
 38 committee. An employee can be a member of multiple committees. Although
 39 already indicated before, I wish to point out that being a member and being the
 40 chairperson of a specific committee exclude each other.
 41 The position of chairperson as well as member of a committee can only be filled
 42 by an employee of our company.
 43 This concludes the information needs regarding project COMMIT.

In fact orientation 102 (FO102), the students learn how to define a conceptual schema using the CSDP (Conceptual Schema Design Procedure) [5] enhanced with integrated concept definitions, three kinds of fact type forms and a few other things, consistently applied to abstract models. An example of an abstract example is given below:

Case CV2

1 A number of engineers have developed a symbolic notation, in order to administer
 2 technical information in a clear and useful manner. The symbolic notation consists
 3 only of circles, rectangles, dashed and solid lines.
 4
 5 Circles and rectangles can only be connected to each other by solid or dashed lines.
 6 Other connections do not exist. It is not allowed to connect two rectangles or two
 7 circles to each other. A circle and a rectangle can be connected to each other only
 8 once, either by a solid or a dashed line.
 9
 10 Each circle is uniquely identified by a number of at most 4 digits, inscribed near the
 11 circle. After every number a plus (+) or a minus (-) is inscribed.
 12
 13 Every rectangle is uniquely identified by a code consisting of 3 characters at most.
 14 At the bottom-left corner of every rectangle some number of at most 3 digits is
 15 inscribed.
 16
 17 A circle can be connected to 0, 1 or more rectangles, of which at most one is con-
 18 nected by a dashed line. Each rectangle is connected by a dashed line to precisely
 19 one circle and also at least once by a solid line to another circle.
 20
 21 The number of rectangles connected by solid lines to the same circle is calculated
 22 for every circle and inscribed within that circle. A comparable number is inscribed
 23 in the upper right corner of every rectangle, namely the calculated number of cir-
 24 cles to which the rectangle is connected by solid lines.

In Fact orientation 103 (FO103), SQL is discussed as a logic language, not from the point of view of programming. In this course, fairly heavy emphasis is put on correlation. All functionality pertaining to specify a result (SELECT), all update operators at ground fact level (INSERT, UPDATE, DELETE) and all schema update operators (CREATE, ALTER, DROP) are covered.

In fact orientation 104 (FO104), the concepts of events and associated (stored) procedures are discussed at the conceptual level, as well as graphical formalisms to express procedures.

In each of these four courses the knowledge triangle is consistently and continuously used as the beacon to which everything is attached. In this way it is possible to relate every aspect of every course to the knowledge triangle. The consequence is that one now has a totally integrated program. The integration of two courses at the domain-specific level is at least via the generic level, but subjects may of course also contain a degree of overlap at the domain-specific level. It is remarkable that persons who have studied SQL are usually not able to decide which knowledge class an SQL query is. In figure 2 the knowledge triangle is illustrated with SQL commands as used in fact orientation 103 (FO103). It can clearly be seen that at Level I, the level of the ground facts, SQL can both read and write. The commands are SELECT for read, and INSERT, UPDATE and DELETE for write. At level II, the domain-specific conceptual schema, SQL can also read and write. The read command is SELECT as it is on Level I, but the write commands are quite different, namely CREATE TABLE, DROP TABLE, ADD COLUMN, ADD CONSTRAINT. At level III SQL can only read, with the SELECT command; SQL can not write to Level III.

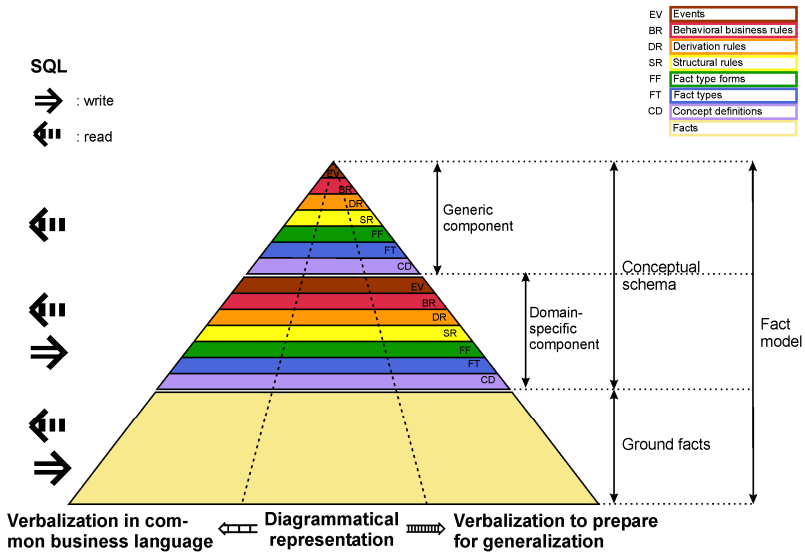


Fig. 2. Knowledge triangle illustrated with SQL

The contents of each of the other 9 courses, the practical work and the thesis are expressed in

- a. a set of concept definitions;
- b. a set of associated fact types;
- c. a set of associated fact type forms;
- d. a set of associated rules.

A course is further illustrated with various “transactions” that show the lifecycle of an element, e.g. the lifecycle of a booking in financial accounting or the lifecycle of a project in the project management subject Prince2; hence extensive use is made of concrete examples, in most cases at the ground fact level.

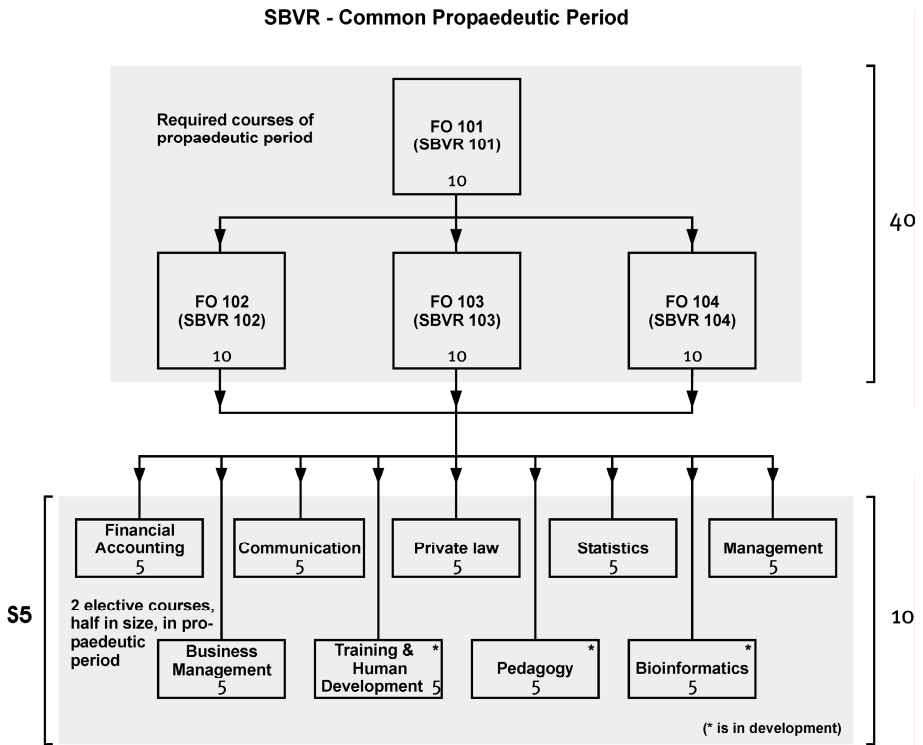


Fig. 3. The structure of the propaedeutic year

In the propaedeutic period there are 4 mandatory courses: FO101, FO102, FO103 and FO104. To avoid any misunderstanding, these four courses have to be taken by every student, regardless of their ultimate specialization. In order to give the student in that period an option to take a closer look at a possible specialization, he can for the fifth course select two courses of half the size. The idea is that a student can then base his selection for the specialization on more experience. He can even select to do 4 more half courses which count as two elective courses in the post-propaedeutic period. In that case the number of students making the wrong selection after the

propaedeutic period, substantially decreases. The specialization Knowledge Management is deemed to be given sufficient attention in FO101 through FO104.

3 Various Specializations

After the 4 initial and mandatory courses of the propaedeutic phase, there are 5 mandatory courses per specialization. This brings the total of mandatory courses per specialization to 9, see figure 1. A student must furthermore select 4 elective courses from a fairly large list of courses, such as financial accounting, project management, UML, logic, communication, didactics, etc.

Although each specialization covers a specific part of Universe of domain-specific worlds, the structure of each is exactly the same as said under figure 1. This gives an increase in the speed of acquiring new knowledge and competencies.

By giving proper attention to semantics, lecturers and students systematically try to connect every new concept to the set of concepts assumed to be known to the student. This results in a level of knowledge and competencies that with the traditional silo education requires several years of working experience.

4 Thesis

For his final thesis the student-employee in most cases specifies – using fact orientation enhanced with concept definitions – the tacit knowledge that he or his colleagues in his department have. In a small number of cases the student selects another subject, e.g., a part of physics or genetics, an area that is not of direct interest to their employer.

At present the required quantities are:

- a. at least 700 concept definitions (assuming the domain-specific level), plus
- b. all associated fact types, fact type forms and rules and
- c. a text that can easily be understood by persons never having been trained in fact orientation.

5 Multi-student Thesis

Some of the more distinguishing interesting aspects of this bachelor program are:

- a. A student can combine one or more practical subjects with his thesis work. This opens the possibility of solving larger problems for the organisation.
- b. Two or more students can combine their practical subjects and theses. This opens entirely new possibilities. However, the coaches of the practical subjects and theses have a much harder task as they must guarantee that each individual student has done his required part of the total work.

The long-term assumption that we want to test is: if we were to subject ‘education’ to a cost-benefit analysis, to which side would the scale tip? If we look at education as making good use of the brains of people to develop new knowledge it may be possible that education could be turned into an indefinitely renewable resource.

6 Examinations

All the examinations are in written format, except for the thesis and the practical subjects.

All written examinations entirely consist of intelligent multiple choice questions. What are intelligent multiple choice questions? A student is given a case study. He has to deliver certain results by applying the competencies he has acquired. He must usually select one of eight available options. A student can consult all his material during the exam (so-called 'open book' exams); he is not permitted to use the telephone, nor the internet.

During the oral thesis and practical subject exams, the student must demonstrate clearly that he is able to give a well-structured presentation and that he is able to answer questions to the satisfaction of the examiners.

7 Experience with the Integrated Program

In most cases the students receive the funding from their employer for the bachelor's degree.

What does the employer get in return for his investment? There are several answers:

- a. The student-employee is more productive in his daily work.
- b. The student employee has developed a much wider view of the organisation, i.e. he has seen that combining the viewpoints of several silos is far more interesting than the often narrow view of one silo.
- c. The student-employee helps the organisation by explicitly delivering a knowledge description of tacit knowledge. We capture the explicit part of tacit knowledge with the use of a universal structure for explicit knowledge (SBVR, CogNIAM). The transfer or training of tacit knowledge is easier and much faster. This means that the organisation is less dependent upon persons with specific tacit knowledge and the knowledge is available to all authorised persons. In a recent thesis of two students at Statistics Netherlands it was computed that the organisation had invested 120.000 Euros in knowledge securing (capturing the tacit knowledge) and exploration; the yearly gains obtained from a much faster introduction of new employees is 200.000 Euros. Most investment bankers would love such a return on investment.
- d. As the student is able to do practical subjects that are of interest to his job he is strongly motivated to spent time on acquiring the competencies. This means that there is no need any longer to send the employee to specialized post-graduate classes as this can be realized in the bachelor education.

What does it do for the employee-student?

- a. He/he has obtained a bachelor degree.
- b. He has experienced that it is possible to integrate all the courses, on the solid foundation of semantic fact orientation. This means that he is intellectually equipped with the knowledge and competencies to perform fully

integrated knowledge work. This integration also includes written and oral communication as well as project management competencies.

- c. The student has experienced that learning to learn based on the knowledge triangle and semantic fact orientation are available today. It is real. Student often indicate that with this approach they finally have learned how to learn very efficiently.
- d. The integration has another pleasant effect and that is that the self-confidence of most students gets a significant boost.
- e. Students obtain during this bachelor program a love for more accuracy.

8 Summary, Expectation and Recommendation

Expressing a course in terms of:

- a. concept definitions
- b. fact types
- c. fact type forms and
- d. rules

makes it possible to make a good step from qualitative thinking in education to quantitative thinking. By further taking into account the domain-specific and generic conceptual schema level, it is possible to develop far more objective yardsticks with respect to subjects or courses. This opens the possibility of managing real results instead of a questionable measure called “contact hours”.

The integration of various subjects in education started in 1989. Step by step the integration has been realized and enlarged to the point where in 2006 an official accreditation was obtained from the Dutch-Flemish Accreditation Council for a fully integrated semantic fact orientation bachelor degree. More than 60 students have finished the bachelor degree in this way.

The expectation of the persons involved in the design and implementation of this bachelor program fully integrated and based on semantic fact orientation is that for cognitive studies sooner or later nearly all such education will become integrated.

The author recommends to all experienced ORM/CogNIAM modelers to extend their interest into other subjects. It is amazing to see what happens if a subject like financial accounting is expressed in SBVR, or statistics, or project management, or UML etc. etc.

References

1. Bollen, P.: Using Fact-Oriented Instructional Design. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1231–1241. Springer, Heidelberg (2006)
2. Chapin, D.: The Essentials of SBVR, Part I and II. The Knowledge Standard, HAN University, Arnhem, The Netherlands (2008)
3. Chapin, D.: SBVR: What is now Possible and Why? *Business Rules Journal* 9(3) (2008), <http://www.BRCommunity.com/a2008/b407.html>

4. Hall, J.: Business Semantics of Business Rules. *Business Rules Journal* 5(3) (2004), <http://www.BRCommunity.com/a2004/b182.html>
5. Halpin, T., Morgan, T.: *Information Modeling and Relational Databases*. Morgan Kaufmann, San Francisco (2008)
6. Nijssen, S.: SBVR: Semantics for Business. *Business Rules Journal* 8(10) (2007), <http://www.BRCommunity.com/a2007/b367.html>
7. Nijssen, S., Bijlsma, R.: A Conceptual Structure of Knowledge as a Basis for Instructional Designs. In: *Proceedings of the Sixth International Conference on Advanced Learning Technologies (ICALT 2006)*, pp. 7–9. IEEE Computer Society, Los Alamitos (2006)
8. NVAO, Accreditation Organization of the Netherlands and Flanders, <http://www.nvao.nl>
9. OMG (Object Management Group), Semantics of Business Vocabulary and Business Rules (SBVR), v1.0. Online as document 08-01-02 (2008), <http://www.omg.org/spec/SBVR/1.0/PDF>. SBVR1.0 supporting files <http://www.omg.org/spec/SBVR/1.0/>
10. Sowa, J.: Fads and Fallacies about Logic. *IEEE Intelligent Systems* 22(2), 84–87 (2007), <http://www.jfsowa.com/pubs/fflogic.htm>
11. NijssenVos, J.: Is There Fact Orientation Life Preceding Requirements? In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM-WS 2007, Part I. LNCS*, vol. 4805, pp. 688–698. Springer, Heidelberg (2007)

SBVR: A Fact-Oriented OMG Standard

Peter Bollen

Department of Organization and Strategy
Faculty of Economics and Business Administration
Maastricht University
P.O.Box 616
6200 md Maastricht, The Netherlands
p.bollen@os.unimaas.nl
Tel.: 31-43-3883715; fax: 31-43-3884893

Abstract. In this paper we will give an introduction to the recently established OMG SBVR standard on business rules. This standard is a major step forward in improving the productivity of business rule- modelers and analysts. The paper furthermore, illustrates how the mature fact-oriented approaches, e.g. ORM and CogNiam, are related to this new standard and how they can contribute to deliver high quality SBVR models.

Keywords: SBVR, Business Rules, ORM, CogNIAM, Fact-Orientation, OMG, NIAM, Semantics of Business Vocabulary and Business Rules.

1 Introduction

Recently, the Object Management Group (OMG) has approved a new standard on business rules: ‘Semantics of Business Vocabulary and Business Rules’ (SBVR). SBVR is a standard for expressing business models for business requirements in the language that is understandable for the business domain users. OMG’s SBVR standard is defined with the aim that business people can understand models without needing IT skills [1]. SBVR defines a structured sub-set of English vocabulary for defining business vocabularies and business rules [2]. SBVR is based upon fact-orientation (ORM/NIAM) and builds strongly on the foundation of ISO-terminology science standards ISO 704: 2000 and ISO 1087-1: 2000 [3] and Linguistics underpinned with formal (first-order) logic [4].

OMG approved the SBVR in September 2005 to become a final adopted OMG specification. In March 2006 the first interim specification was issued [5]. Finally, on December 11, 2007 SBVR became an official OMG specification [6].

In this article we will introduce the main building blocks of the SBVR and we will compare the definitions of a number of modeling elements to ‘familiar’ modeling concepts in ORM [7] and CogNIAM [8-10]. In section 2 a short introduction to ORM/CogNIAM will be given. In section 3 we will discuss the main building blocks of the SBVR. In section 4 we will see how the knowledge modeling procedure in ORM/CogNIAM can be used to generate complete SBVR models. Finally, in section 5 conclusions will be drawn.

2 Introduction to ORM/CogNIAM

ORM [7] and CogNIAM [9, 10] are members of a family of conceptual modeling approaches that traditionally have been used for the specification of relational databases [7, 11] but have evolved into business rule modeling languages [12-16] and languages for subject matter modeling [17-20]. We will refer to this family of approaches as fact-oriented modeling approaches. The fact-oriented approach structures verbalizable knowledge into the following elements [17]:

1. Knowledge domain sentences (or fact instances).
2. Concept definitions and naming conventions (or reference schemes) for concepts used in domain sentences.
3. Fact types.
4. Fact type readings for the fact types.
5. Population state (transition) constraints for the knowledge domain.
6. Derivation rules that specify *how* specific domain sentences can be derived from other domain sentences.
7. Exchange rules that specify *what* fact instances can be inserted, updated or deleted.
8. Event rules that specify *when* a fact is derived from other facts or when (a) fact (s) must be inserted, updated or deleted.

The combined Knowledge Reference Model (KRM) consisting of elements 1 through 8 of the above captures a complete description of a domain- or application's conceptual schema, including the domain- or application ontology.

3 The Building Blocks of SBVR Models

The SBVR is applied within the general notion of OMG's model-driven architecture (MDA) and is targeted at business rules, business concepts and business vocabularies that describe businesses themselves rather than the possible IT system that might support it. The 5 most important aspects in the SBVR can be considered [6, pp. 224-225, 21]:

1. **Formal (first-order predicate) logic**,
2. The recognition of the existences of a semantic community united by a body of shared meanings, possibly having multiple user (speech) **sub-communities** having their own languages and specialized vocabularies,
3. A **body of shared meanings**, represented in concepts, fact types, and business rules for these *sub-communities* underpinned by *formal logic*,
4. A **logical formulation**, for capturing the semantics of *a body of shared meanings*, that supports multiple forms of representation and is underpinned by *formal logic*.
5. A **business representation** for the *logical formulation* of semantics using vocabularies acceptable to the (speech) *sub-community*.

In this paper we will focus on those elements in OMG's SBVR standard that refer to meaning or semantics. The main building blocks for semantic in the SBVR are the

following: *vocabularies and terminology dictionaries, noun- and verb concepts, and definitional- and operational business rules.*

In this paper we will illustrate the definitions in the standard by referencing example applications in the EU-rent case study that is attached to the standard document as Annex E [6, pp. 267-342].

3.1 Vocabularies and Terminology Dictionaries

One of the new features that has been introduced by the SBVR to the field of conceptual business modeling at large is the explicit definition of (external) vocabularies and namespaces. This allows to qualify signifiers by adding the name of the applicable context vocabulary (e.g., car rental industry standard glossary). In SBVR the applicable context vocabularies are defined as speech communities and vocabularies [6, p.274-275]:

' Car Rental Industry Standard Glossary

Definition: the vocabulary that is defined in English by the *Car Rental Industry*

Synonym: *CRISG*

Reference Scheme: *CRISG* terms

CRISG

Synonym: *Car Rental Industry Standard Glossary*

Merriam-Webster Unabridged Dictionary

Definition: the vocabulary that is the 2004 edition, published by Merriam-Webster

Synonym: *MWU*

Reference Scheme: *MWU* terms.

MWU

Synonym: *Merriam-Webster Unabridged Dictionary* '

3.2 Noun- and Verb Concepts

An explicit modeling assumption (or axiom) in the SBVR standard is the reference to *facts* and *terms*, respectively: 'rules are based on facts, and facts are based on terms' [6, p.234]. This 'mantra', implies at least a 'way of working' in which (verbalized) concepts are defined, before fact type (forms) can be phrased. Therefore we need to find (a) the fact type(s) for every business rule that needs to be modeled.

Noun Concepts in SBVR. In the SBVR 1.0 specification a noun concept is defined as a 'concept that is the meaning of a noun or noun phrase' [6, p.19]. An object type is defined as follows: 'noun concept that classifies things on the basis of their common properties' [6, p.19]. Role is defined as: 'noun concept that corresponds to things based on their playing a part, assuming a function or being used in some situation' [6, p.20]. An individual concept is as: ' a (noun) concept that corresponds to only one object [thing]' [6, p.21]. In paragraph 1 of clause 8 of the SBVR 1.0 standard document [6, pp. 19-25] it is clearly explained that the noun concept has as subtypes: individual concept, object type and fact type role.

Verb Concepts in SBVR. In the SBVR 1.0 specification ‘verb-concept’ is synonym for ‘fact type’ and is defined as follows: ‘a concept that is the meaning of a verb phrase that involves one or more noun concepts and whose instances are all actualities.’[6, p.21, 183]. An example of an expression of a verb-concept or fact type expressed in SBVR-structured english is the following:

rental car is stored at branch

SBVR does not contain an ‘attribute’ fact encoding construct as is the case in most non-fact oriented modeling languages like UML and (E)ER and therefore, SBVR prevents the associated modeling anomalies, that can occur when the attribute modeling construct is applied [22]. The SBVR fact type definition is as follows [6, p.21]:

<p>Fact type</p> <p>Definition: <u>concept</u> that is the meaning of a verb phrase that involves one or more noun concepts and whose instances are all actualities.</p> <p>Synonym: <u>verb concept</u></p> <p>.....</p> <p>Necessity: Each <u>fact type</u> <i>has</i> at least one <u>role</u> ‘</p>
--

The above definition fragment, clearly demonstrates that the basic fact type definition in the SBVR is a fact-oriented definition comparable to ORM/CogNIAM and allows for fact types having arity N !. Furthermore, special definitions are provided for unary fact types (or characteristics) and binary fact types. From this it follows that SBVR is a fact-oriented business rule standard.

Fact Type Forms. A designation that represents a fact type in SBVR is demonstrated by a fact type form. A fact type form contains a fact type reading that includes placeholders. This implements ORM/CogNIAM’s fact type template and placeholder.

3.3 Types of Business Rules in SBVR

The most common way of initially expressing business rules in SBVR is by means of a subset of the English Language : SBVR’s structured english [6, Annex C]. An example of a rule expression in SBVR structured English is the following:

each rental car is stored at at most one branch.

In this example we have two designations for an object type: *rental car* and *branch*. Furthermore, we have the quantifiers: *each* and *at most one*. Clause 12 of SBVR v 1.0 [6, pp. 157-177] covers the definition of the types of business statements that can be distinguished in a given business domain. The main types of rule statements are the *structural business rule* statement and the *operative rule* statement. Within each of these groups, SBVR uses two styles of keywords for expressing the business rule statements.

Structural (or Definitional) Business Rules. In the SBVR 1.0 specification, a structural rule is defined as: a rule that is a claim of necessity [6, p.161]. A structural

business rule statement can take one of the following forms: necessity business rule statement, impossibility business rule statement, restricted possibility rule statement.

A necessity statement is defined : ‘.. as a structural rule statement that is expressed positively in terms of necessity rather than negatively in terms of impossibility.’ [6, p. 168]. An example of an structural business rule expressed as a necessity business rule statement in pre-fixed style is:

‘**It is necessary** that *each* rental has *exactly one* requested car group.’

We note that in the above necessity business rule statement, we have put in italics, the quantification keywords *each* and *exactly one*. An example of a structural business rule expressed in a impossibility business rule statement in pre-fix style is:

‘**It is impossible that** the pick-up branch of a one-way rental is the return branch of that rental.’

A structural business rule expressed as a pre-fix restricted possibility statement is the following:

‘**It is possible that** a rental is an open rental only if the rental car of the rental has been picked up.’

The structural business rules in SBVR are so-called alethic constraints, that are true by definition and therefore cannot be violated by the business.

Our example fact type and the example business rule (see section 3.3.4) are expressed in SBVR using the following SBVR expressions [6, p.316]:

‘ rental car *is stored* at branch
Necessity: Each rental car is stored at most one branch ‘

Operative (or Behavioural) Business Rules. In the SBVR 1.0 specification an operative business rule is defined as follows: ‘..business rule that is a claim of obligation’ [6, p.161]. An operative business rule is expressed in SBVR as an operative business rule statement, that can take one of the following forms: obligation statement, prohibitive statement and restricted permissive statement. An example of an operative business rule expressed in an obligation statement in an embedded style [23] is:

‘A rental **must** incur a location penalty charge if the drop-off location of the rental is not the EU-Rent site of the return branch of the rental.’

An example of an operative business rule expressed in a prohibitive statement is:

‘A rental **must not** be open if a driver of the rental is a barred driver.’

An operative business rule expressed as a restrictive permissive statement is the following:

‘ **It is permitted that** a rental is open only if an estimated rental charge is provisionally charged to the credit card of the renter of the rental.’

An operative business rule is actionable, but not necessarily automatable, it can therefore be broken by people. The existence of operative business rules or deontic constraints, furthermore, allows the SBVR to document work-instructions and other rules of guidance, that have been traditionally outside the scope of traditional languages for (conceptual) business modeling.

Table 1. Rule templates in SBVR for keyword style/rule type combinations

Modality Type	Prefixed Style	Embedded Style
<i>Definitional/Structural</i>		
Necessity	It is necessary that	...always...
Impossibility	It is impossible that	...never...
Restricted possibility	It is possible that	...sometimes...
<i>Operative/behavioural</i>		
Obligation	It is obligatory thatmust...
Prohibition	It is prohibited thatmust not...
Restricted permission	It is permitted thatmay...

The SBVR standard has been created to help the business to model explicit (enforceable at all times) rules as well as tacit rules (in which the action that has to be undertaken depends upon for example the experience of a business user) and the (static) alethic and deontic constraints that exist in the business domain. In sub-clause 10.1.1.4 through 10.1.1.6 of the SBVR 1.0 standard [6, pp. 97- 107], the semantic and logic foundation for the standard are provided, in which the formal equivalence between each of the 3 forms for each of the two statement types is provided. In table 1, the rule templates are provided for each style/rule type combination [6, p. 230-231, p.345, 23].

4 A Methodology to Define the Complete SBVR Model for an Application Subject Area

In this section we will illustrate how we can derive a textual SBVR business rule specification that contains a complete *domain vocabulary*, a complete description of *fact types* and a complete set of *structural-* and *operative business rules*.

There are two archetypical ways of finding (structural) business rules. The first way of doing it is asking domain experts, to phrase business rules out of the blue. The second way of finding business rules, is by means of some ‘procedure’ or ‘cook-book’ in which a business analyst guides a business domain user in eliciting the relevant business rules in a dialogue in which a user is confronted with examples, leading to a set of answers that can be translated into domain business rules in combination with the business analyst. The SBVR standard, itself does not discuss the modeling methodology issues, explicitly, but in the main body of the specification some kind of ‘in-between’ approach is suggested, in which the availability of a ‘fact type’ statement is provided as a base for phrasing structural and/or behavioural business rules. Fortunately, in Annex L of the standard document [6, pp.397- 403], CogNIAM’s knowledge modeling procedure is given that specifies how a conceptual schema can be created and in which a ‘cook-book’ for deriving uniqueness constraints, is precisely described. We will now apply ORM/CogNIAM’s modeling procedure on our scaled-down EU-rent example.

4.1 Classifying, Qualifying and Creating the List of Concept Definitions

We will now extend our EU-rent example, by taking the example document or ‘data use case’ in figure 1 as a starting point.

Rental car	Branch	Fuel level
VIN09	Maastricht	
VIN08	Sittard	7/8
VIN03	Sittard	
VIN92	Eindhoven	Empty

Fig. 1. Example of communications EU-rent

The initial verbalization of the example of communication in figure 1 leads (amongst others), to the following sentences:

- ‘The rental car VIN09 is stored at branch Maastricht’
- ‘The rental VIN08 is stored at branch Sittard’
- ‘The rental car VIN92 has fuel level Empty’
- ‘The rental car VIN92 is stored at branch Eindhoven’

Now we have verbalized the example of communication, we can qualify the ‘variable’ positions by finding the name classes and by phrasing ‘naming-convention’ facts [24, p. 134]:

- ‘Within the class of all rental cars of EU-rent the vehicle identification number VIN09 identifies a specific rental car.’
- ‘Within the class of all branches of EU-rent the name Sittard identifies a specific branch.’

In table 2 we have shown CogNIAM’s list of concept definitions, in which we have defined the object types and name classes that were ‘discovered’ after the verbalization of the ‘familiar’ example, in order of comprehension.

Table 2. CogNIAMs List of concept definitions in order of comprehension

Concept	Definition
Rental Car	vehicle owned by EU-rent and rented to its customers <u>Synonym:</u> car
Vehicle identification number	text that is the unique identifier of a particular [rental car]
Rental organization unit	organization unit that operates part of EU-Rent’s car rental business
Branch	[rental organization unit] that has rental responsibility
Name	identifier of a particular [branch]
Fuel level	The relative content of the fuel tank of a [rental car]
Rental Car is stored at Branch	A [rental car] must have a place where it can be physically stored until it is rented to customer

4.2 Deriving Uniqueness, Mandatory Role- and Value-Constraints Using ORM/CogNIAM

The example fact types expressed in SBVR-structured English:

- rental car is stored at branch
- rental car has fuel level

and the accompanying constraints or ‘business rules’ are:

- each rental car is stored at at most one branch.
- a rental car has at most one fuel level.

The first structured English business rule can be stated as an impossibility structural rule statement in SBVR as follows:

‘It is impossible that a rental car is stored at more than one branch’

or as a SBVR business rule expressed as a necessity business rule statement:

‘It is necessary that *each* rental car *is stored at* at most one branch.’

or as a SBVR business rule expressed as a restricted possibility rule statement:

‘It is possible that a rental car *is stored in* at most one branch’

The above SBVR business rule is depicted in figure 2 as ORM/CogNIAM constraint *uc1*. If we inspect figure 2 further, we see that the application of the ORM/CogNIAM modeling procedure has resulted in the detection of uniqueness constraint *uc2*, which was not listed in the EU rent SBVR model in appendix E in [6] but should have been phrased as the following structural business rule in SBVR:

‘It is impossible that a rental car has more than one fuel level’

In figure 2 we have given the extended fact diagram (based upon parts of the section rental cars in the EU rent-example) and a significant population and the outcome of the ORM/CogNIAM mandatory role constraint derivation procedure: mandatory role constraint *mc1*. This constraint can be stated as a necessity rule statement in SBVR as follows:

‘It is necessary that *a* rental car *is stored at* a branch.’

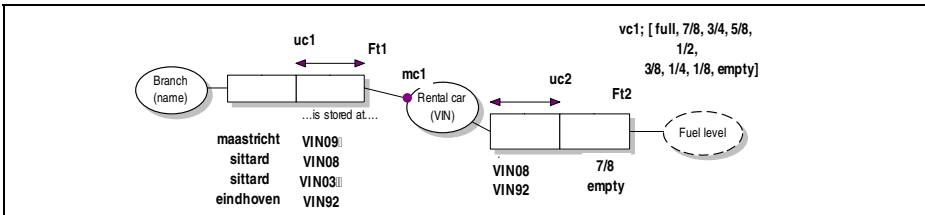


Fig. 2. ORM diagram for example fact type and mandatory role constraints

The last pre-defined ORM/CogNIAM constraint type we will derive is the *value constraint type*. We note that in the SBVR specification’s EU-rent example [6, p. 315], the concept of *fuel level* is defined as a set of value instances. In ORM/CogNIAM, this can be mapped as a value type plus the value constraint *vc1*.

4.3 Mapping ORM/CogNIAM Conceptual Models onto SBVR Vocabularies

The application of the ORM conceptual schema design procedure [7] based on accepting/rejecting combinations of ground facts provided by the analyst to a domain expert leads to set of uniqueness and mandatory role and value constraints that can be mapped onto elementary SBVR rule statements. Because the SBVR is founded (amongst other things) on the essential concepts in fact-orientation, first-order logic and terminology science, it is possible to create a language processor [25] that produces SBVR models out of an ORM/CogNIAM model. On the other hand the current SBVR standard does not give guidance on how to present the elements in an SBVR specification, other than the form in which the ‘SBVR meta’ specification is presented

and the way in which the EU-rent case study is presented: mostly textual and rule statements and fact types are dispersed and sometimes overlapping. Some SBVR practitioners propose to express SBVR specifications in a more ‘user-friendly’ way, for example, by presenting business rule statements ‘around’ the fact type roles, that are involved in such a business rule [26]. Exchange rules and events, for now are not defined in the SBVR standard. In table 3 we have given an example of a possible SBVR compliant vocabulary for our scaled-down EU-rent example that can be fully generated from an ORM/CogNIAM knowledge structure model.

Table 3. SBVR compliant vocabulary of our example subset of EU-rent

<u>rental car</u>	Definition: vehicle owned by EU-rent and rented to its customers Synonym: <u>car</u>
<u>vehicle identification number</u>	Concept type: <u>role</u> Definition: text that is the unique identifier of a particular vehicle Synonym: <u>VIN</u>
<u>rental car</u> <i>has</i> <u>vehicle identification number</u>	Necessity: Each rental car <i>has</i> exactly one vehicle identification number.
<u>rental organization unit</u>	Concept type: <u>role</u> Definition: organization unit that operates part of EU-Rent’s car rental business
<u>branch</u>	Definition: <u>rental organization unit</u> that has rental responsibility Necessity: the <u>concept branch</u> is included in <u>organization units by function</u>
<u>branch</u> <i>has</i> <u>name</u>	Concept type: <u>is-property-of fact type</u>
<u>fuel level</u>	Definition: <u>full</u> or <u>7/8</u> or <u>3/4</u> or <u>5/8</u> or <u>1/2</u> or <u>3/8</u> or <u>1/4</u> or <u>1/8</u> or <u>empty</u>
<u>rental car</u> <i>has</i> <u>fuel level</u>	Necessity: Each <u>rental car</u> <i>has</i> at most one <u>fuel level</u>
<u>rental car</u> <i>is stored</i> at <u>branch</u>	Necessity: Each <u>rental car</u> <i>is stored</i> at exactly one <u>branch</u>

5 Conclusions

The SBVR standard gives us the modeling concepts to define most, if not all business rules that can be encountered within organizations. The establishment of an OMG standard for the semantic vocabulary of business rules has been a major step forward in the process of making business domain knowledge explicit and transferable. The SBVR does not, however, give a procedure or methodology on how to arrive at complete and correct SBVR models. We recommend, however, to conceptually model a business domain using ORM/CogNIAM’s modeling methodology as is illustrated in Annexes I and L of the SBVR v1.0 standard document as a first step. This enables us to capitalize on the verbalization of user examples and the subsequent permutations of example populations to arrive at the underlying ORM/CogNIAM population constraints. Subsequently, the found list of concept definitions, fact type(s) (readings),

population constraints and derivation rules can be easily mapped onto a SBVR compliant vocabulary and a compliant set of SBVR business rule statements.

References

1. Nijssen, G.: Hooray, SBVR has arrived! *Business Rules Journal* 9(2) (2008)
2. Vanthienen, J.: The ABCs of Accurate Business. *Business Rules Journal*, 9(3) (2008)
3. Baisley, D., Hall, J., Chapin, D.: *Semantic Formulations in SBVR* (2005)
4. Chapin, D.: SBVR: What is now possible and why? *Bus. Rules Journal*, 9(3) (2008)
5. OMG, SBVR, first interim specification, p. 392 (2006)
6. OMG, *Semantics of Business Vocabulary and Business Rules (SBVR)*, v1.0 (2008)
7. Halpin, T., Morgan, T.: *Information Modeling and Relational Databases*, 2nd edn. Morgan-Kaufman, San-Francisco (2008)
8. Nijssen, G.: CMMI, SBVR en UML. Een totale ontwikkelingsaanpak. In: 1st SBVR foundation conference, Utrecht, the netherlands (2008)
9. Lemmens, I., Nijssen, M., Nijssen, G.: A NIAM 2007 conceptual analysis of the ISO and OMG MOF four layer metadata architectures. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM-WS 2007, Part I. LNCS*, vol. 4805, pp. 613–623. Springer, Heidelberg (2007)
10. Nijssen, G.: CMMI, SBVR en UML: een totale ontwikkelingsaanpak. In: 1st SBVR foundation conference, Utrecht (2008)
11. Nijssen, G., Halpin, T.: *Conceptual schema and relational database design: a fact oriented approach*, p. 337. Prentice-hall, New-York (1989)
12. Nijssen, G.: *Semantics for Business: a process to specify the most important business rule in SVBR*. *Business Rules Journal*, 8(12) (2007)
13. Nijssen, G.: *SBVR: Semantics for Business*. *Business Rules Journal*, 8(10) (2007)
14. Nijssen, G.: *Kenniskunde 1A*. PNA Publishing Heerlen (2001)
15. Halpin, T.: *Business Rule Modality*. In: *Proc. EMMSAD 2006: 11th Int. IFIP WG8.1 Workshop on Exploring Modeling Methods in Systems Analysis and Design* (2006)
16. Halpin, T.: *A fact-oriented approach to business rules*. In: Laender, A.H.F., Liddle, S.W., Storey, V.C. (eds.) *ER 2000. LNCS*, vol. 1920, pp. 582–583. Springer, Heidelberg (2000)
17. Nijssen, G., Bijlsma, R.: *A conceptual structure of knowledge as a basis for instructional designs*. In: *The 6th IEEE international conference on Advanced Learning Technologies, ICALT 2006, Kerkrade, the Netherlands* (2006)
18. Bollen, P.: *Using Fact-Oriented for Instructional Design*. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops. LNCS*, vol. 4278, pp. 1231–1241. Springer, Heidelberg (2006)
19. Vos, J.: *Is there fact orientation life preceding requirements?* In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM-WS 2007, Part I. LNCS*, vol. 4805, pp. 688–698. Springer, Heidelberg (2007)
20. Nijssen, G., Bollen, P.: *Universal Learning: a science and methodology for education and training*. In: *EDINEB: the case of problem-based learning*. Kluwer, Dordrecht (1995)
21. *SBVR Excerpt, The five major aspects of SBVR*. *Business Rules Journal*, 7(1) (2006)
22. Bollen, P.: *How to overcome pitfalls of (E)ER and UML in Knowledge Management education*. *Int. Journal of Teaching and Case Studies* 1(3), 200–223 (2008)
23. *SBVR Excerpt, Notations for business rule expressions*. *Bus. Rules J* 7(4) (2006)
24. Bollen, P.: *The Natural Language Modeling Procedure*. In: *5th international workshop on Next Generation Information Techn. and Syst. 2002*. Springer, Caesarea (2002)
25. Baisley, D.: *SBVR: What are the Possibilities?* *Business Rules Journal*, 9(3) (2008)
26. Nijssen, G., Hall, J.: *SBVR diagram; a response to an invitation*. *Business Rules Journal*, 9(7) (2008)

An Adaptable ORM Metamodel to Support Traceability of Business Requirements across System Development Life Cycle Phases

Baba Piprani¹, Marlena Borg², Josée Chabot², and Éric Chartrand²

¹ SICOM Canada

² Transport Canada

`babap@attglobal.net`, `{BORGME, CHABJOS, CHARTRE}@tc.gc.ca`

Abstract. Enterprises launching IT development projects usually start off with establishing Use Cases or similar techniques to document functional requirements as a special directed effort. More often than not, the resulting information system has buried these and other newly discovered undocumented requirements into program code--losing the important link between business requirements, business rules and developed code. In reality, the business requirements are generally surfaced over several years in memos, e-mails, meeting minutes, consultant reports etc., and the requirements gathering effort starts all over again as a new project to capture these already stated requirements. Using an ORM based development life cycle approach, the supporting adaptable traceability metamodel enables the collection and tagging of the business requirements across a multitude of documents and across development phases, to provide traceability and the facility to develop transforms across an organized information system development effort in meeting with any established System Development Life Cycles.

Keywords: Requirements traceability, SDLC, ORM, Business requirements, metamodel.

1 Why Requirements Traceability?

When people leave organizations, more often than not, corporate memory leaves with them. The remaining staff may not remember the need for a particular requirement or business rule, why it may have changed or why it may no longer be required.

Some systems take years to come into being for a variety of reasons. There may be lengthy lapses of time between life cycle phases due to higher priority projects, lack of resources, project staff changes, etc.

A requirements traceability metamodel would reduce the possibility of losing track of the requirement throughout the life cycle. It would also provide a more systematic way to analyze business requirements and rules, identify duplicate or conflicting requirements and rules, and reduce the possibility of conflicting results since every requirement is easily traceable and can be tracked.

Another important set of deliverables from a requirements traceability metamodel would be that it provides the capability to follow the lineage of the requirement and to ensure that the necessary business requirements are captured at a point in time; are updated as required while keeping a history and rationale for changes; and are definitively addressed somewhere in some phase of the system development life cycle.

There are several benefits from establishing the requirements traceability metamodel.

For the Organization:

- provide continuity, promotes good governance of data and protection of corporate memory.
- increased stakeholder confidence in the organization with respect to products offered.
- increased productivity of project staff .

For the Business Client (or stakeholder):

- eliminate or reduce time and effort to continually explain and rationalize requirements.
- improve confidence in the system development process and that the final product will be fully reflective of their needs.
- improve confidence in system output.

For the System Developer:

- requirements and business rules would be documented in one place reducing the time and effort required to analyze and track the requirements.
- should project staff change, or should there be a lengthy lapse of time between phases, it would eliminate the need to repeat locating, analyzing and confirming requirements.
- duplicate or conflicting requirements would be more evident, thus reducing analysis effort and the risk of overlooking such duplications or conflicts at an early stage of the development process.
- facilitate and accelerate verification that all requirements and rules have been addressed by the system (i.e. improved quality assurance).

The focus of this paper is to provide an ORM schema and its associated attribute based relational schema depicting an implementation of a requirements traceability metamodel, and explain its usage in a real-life scenario.

This paper addresses the need for requirements traceability, a high level overview of the currently used ORM based system development life cycle, the ORM metamodel as used for business requirements tracking and as implemented, some usage scenarios, and the mapping to the corporate development life cycle model.

2 An ORM Based System Development Life Cycle

Figure 1 below depicts a typical ORM based System Development Life Cycle initiated by propositions involving business requirement statements, that incorporates

multiple phases of analyses incorporating Business Activity Modelling (IDEF0[1]), Process Modelling (BPMN[2] or BPWin and IDEF3[3]), ORM Modelling, Attribute Based Modelling (IDEF1X[4], ERWin), Event Modelling, Control Sequence Modelling, and being realized or implemented in a Service Oriented Architecture using an Oracle physical schema and supporting BI toolsets like Crystal Reports, Business Objects---using the standard data modelling, process modelling and implementation tool sets at Transport Canada.

The following paragraphs briefly describe a summary description of each phase and the identifying concepts that are involved in the business requirements traceability for the purpose of the metamodel as depicted in this paper.

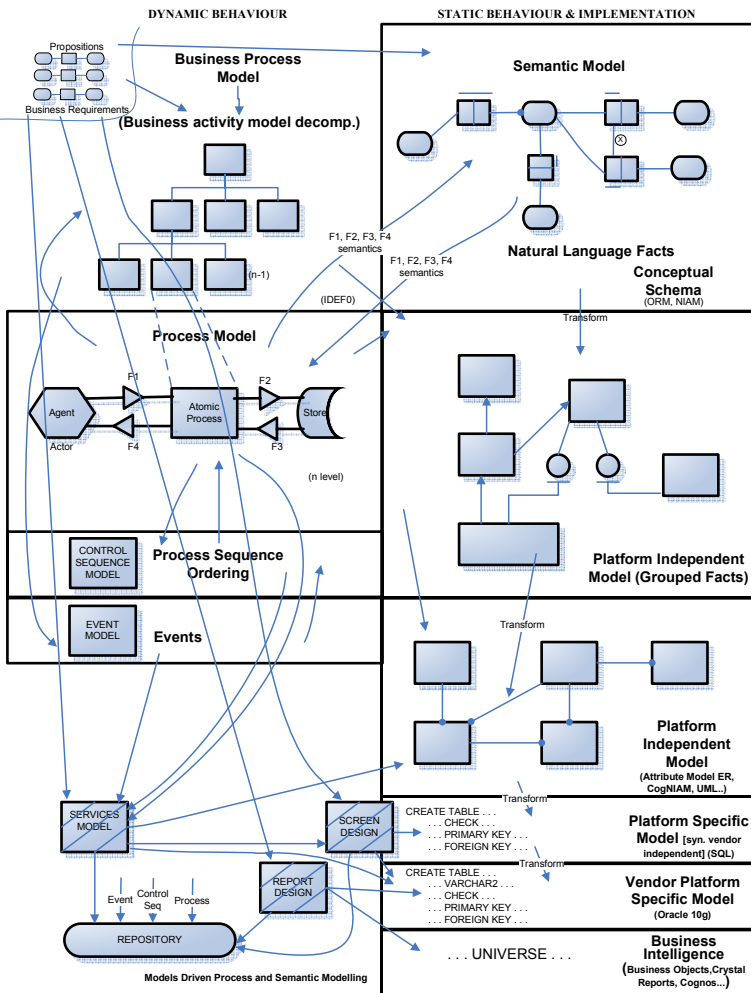


Fig. 1. An ORM based System Development Life Cycle at Transport Canada

As can be seen from Figure 1, the progression of systems development is not a waterfall approach, nor a spiral discover-as-you-go approach, but instead, a services based architecture that has its analysis taking place in a zig-zag fashion, pretty much mimicking the way a human brain thinks. Humans tend to multi-thread and connect several frames and concepts in their thought processes. Similarly, the analysis exercise as depicted essentially is initiated from a statement of business requirements.

2.1 The Business Activity Model to the Semantic Modelling Phase

An initial forest-level picture pegging the boundaries of the set of stated requirements is defined through a form of functional business decomposition. A business activity contributes to the achievement of an objective of the business. Each business activity in the hierarchy is decomposed until the lowest level activities (called elementary business activities or atomic processes---cannot be split further) that comprise it have been identified. This lowest level activity may contain a mix of automatable and non-automatable activities where the automatable part cannot be split any further without encountering database input / output operations like Create/Retrieve/Update/Delete, or, purely an automatable process that again encounters database input / output operations like Create/Retrieve/Update/Delete, or lastly, a purely business non-automatable process that would simply contain a sequence of operations that are external to the automatable domain.

This last (or lowest) level is denoted as level "n", i.e. an automatable process. The business decomposition stops at level "n-1", i.e. the level when the activity involves an "automatable part" and still maintaining a "purely business part", and where a further decomposition results in a purely automatable process involving computer facilities input / output, or, the consumption or generation of a product. The lowest level n of a decomposition becomes a process that involves some form of input or output in an automated data processing system.

The procedures for defining business activities and the procedures for decomposition may be done differently by different people. In other words, two or more persons defining business activities and decompositions of the same business may arrive at different answers. This is quite acceptable and it does not matter, as long as all the business activities are being covered.

Why does this not matter? A business activity model is not a formal model i.e. there is not a formal grammar to support the business activity model. What matters is the data or information that is to be identified and formalized in the data usages of information flows from the lowest level process.

It is important that the lowest atomic processes represent a complete elementary task activity that cannot be split any further without losing meaning, and that these elementary tasks, while they may contain a processing sequence to accomplish that elementary task, may not be connected or sequenced with other elementary tasks---since this sequencing actually is a service and is depicted by an independent stand-alone sequence model that controls the sequence of atomic processes.

A semantic data model is derived from the data usages in the information flows of these atomic processes. A semantic data model is a formal model with formal grammar associated with it.

What this means, is that it does not matter how the business activities are organized, as long as the data usages have been recorded. No matter which alternate approaches of business activity modelling or decomposition is used---be it organizational based, product association based, business functionality based---the data usage information flows from the lowest level processes will ultimately result in a single verifiable formal data grammar / semantic schema. This is because the final implementation is essentially being supported by a Services Model to achieve the business deliverables of the enterprise based on these agreed upon semantics. The decomposition of business activities is only a means of achieving the formalization of the semantic data model required to support the enterprise. It is the Services Model that will bring the necessary atomic processes, their necessary sequences along with pre and post conditions to enable the carrying out of the necessary services for the enterprise as derived from the requirements.

2.2 Transforms from Semantic Data Model to Neutral and Physical Data Model

It is this ORM schema that is used as the foundation for the development of an attribute based model (IDEFIX in ERWin) and further transforms to the neutral data model, and physical data model (Oracle SQL). These are generally published mappings provided by tool manufacturers or other previously published materials.

It is important to note that the data model transforms need to carry with them the maximal scope of integrity rules and constraints.

While the initial scope of the semantic data model was being defined by the data usage information flows of each elementary business activity or atomic process, one will note that this “Process Model connection” appears to be “left behind” over the transforms to the physical SQL schema.

2.3 Bringing the Processes Together

Recall that the elementary business activity or an atomic process, while it may have its own internal sequence to complete the elementary task (e.g. change reservation date of hotel guest), it should not be associated with another process in any sequence except if that process is calling another process to complete its task. For example the atomic process “change reservation date of hotel guest” will require a re-usable atomic process “select hotel guest folder” which will simply fetch the current reservation and other account details of the given hotel guest.

The sequence of processes to be performed is determined by a Services Model which has a set of processes that is driven by events and in turn uses a control sequence model that determines which process is to be performed for that particular service.

Of course, the processes may require data from multiple database sources or URIs.

3 Harmonizing the Business Requirements to the SDLC Model Suite

Many System Development Life Cycles (SDLC) [5] typically have a separate requirements collection phase along with identified deliverables along the way, and many embark upon this “Business Requirements Collection” journey as a search for

the Holy Grail, hoping to receive a “Wal-Mart package style” set of requirements on a plate. While this is generally a useful exercise, it must be noted that the users are essentially visualizing some services-process based on a business requirement, at the root of which is a data model that can be formally defined and supported by a set of semantic models and associated supporting models.

Enterprises launching IT development projects usually start off with establishing Use Cases that determine system behaviour or similar techniques to document functional requirements as a special directed effort. More often than not, the resulting information system has buried these business requirements and other newly discovered undocumented requirements into program code, losing the important link between functional requirements, business rules and developed code---where business rules can be defined as being conformance requirements for the operations, definitions and constraints that apply to an organization in achieving its goals. In reality, the business requirements are generally surfaced over several years in memos, e-mails, meeting minutes, consultant reports etc., and the requirements gathering effort starts all over again as a new project to capture these already stated requirements.

So the question is, a) how to relate these business requirements collected over time---no matter how assembled---to the various multitude of models, and, b) how to track and define an updated lineage that is easy to trace and maintain?

● **The Business Requirements Metamodel**

Figure 2 depicts an ORM metamodel for business requirements tracking as has been implemented at Transport Canada in the Inspection Information System for the Transportation of Dangerous Goods.

Business requirements have accumulated over a number of years in the form of e-mails, meeting minutes, memos, reports, consultant studies etc.---with each set represented as a separate ‘document’ for ease of referencibility. A document is identified by a unique document identifier. A document may contain one or more requirements, each identified by a uniquely identified sequence within the document. A document may have a document current or past date or is given an approximate current or past date of being recorded. A document must have one or more authors. A document must have a unique title, and may have notes associated with it. A requirement must be identified by the originating document and a sequence number within that document. A requirement as noted in a document is re-worded as a statement of requirement, i.e. as a proposition and noted as a requirement statement description, while the original text is maintained via a document reference.

Admittedly, the requirement statement as extracted may appear to be loosely represented, and not formal since the requirement could span several areas like User Interfaces, business rules, access requirements etc. As such, a requirement here is a summarization of an interpretation of a user communication or request, which is to be verified by the user as representative of the original statement of requirements.

The requirement must be associated with one or more requirement categories. A requirement may also have a requirement status which is determined after examining the requirement. A requirement if duplicated, is addressed as contained in only in one parent requirement. Similarly, a component may be contained in another component and is qualified as such.

In the actual implementation, this requirement status is in the form of a work-flow showing the temporal progression and history of the status of the requirement, but is not depicted in the metamodel shown for the sake of simplicity. Decisions are then made on whether the requirement is to be implemented at all, and to be realized in what implementation component. For example, a requirement could be to address navigation on a screen to enable an inspector to be able to examine the prior inspection history of an organization while conducting a current investigation. This clearly belongs in the screen model. Another requirement could be related strictly to a business fact, for example, the requirement for a ticket in addition to the current set of documents for an inspection (new fact type, new constraint) or relating an investigation across all the involved organizations (new fact type), or a report chart of inspections by region over the past 5 years vs. compliance infractions (derived fact type, report).

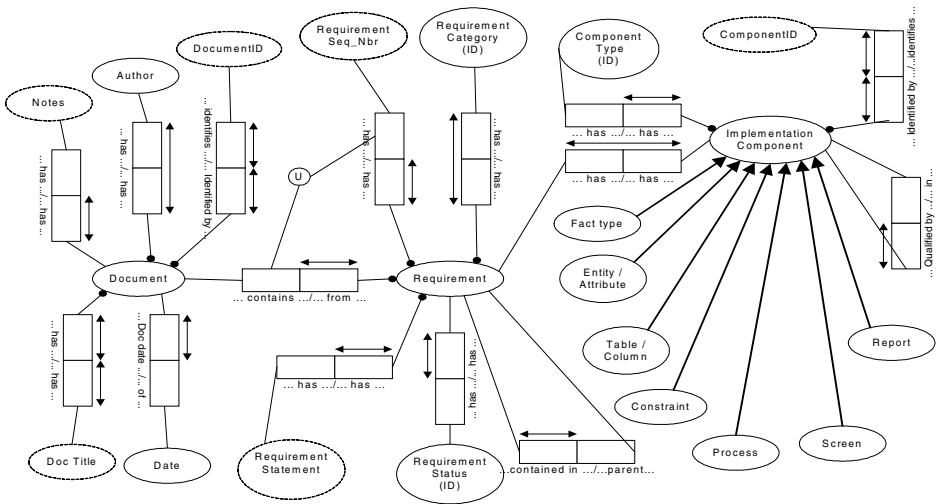


Fig. 2. ORM Metamodel for Business Requirements Tracking

Transforms from requirements to Implementation components are defined at the most primitive level and not to all possible associations. This is to avoid any redundancies. For example, a requirement can be directly implemented via a fact type, which then is mapped through standard transforms to an Entity Attribute data model, which in turn is mapped through standard transforms to an SQL column and table, which is then associated with a process which in turn is within a service offering, etc. In this case, the transform is only maintained to the first primitive association i.e. the fact type and not to the tables, columns, constraints etc. There could be other transforms that may be directly mappable to a set of tables and columns, e.g. the Z900 series of tables (see Figure 1) which are essentially an extended Information Schema tables containing metamodels to track implementation artefacts like which screen uses which column and which report uses which column, which process belongs to which service etc.

The ORM metamodel for business requirements tracking as transformed to the ER attribute based model using IDEF1X in ERWin--the Transport Canada Data Modeling standard toolset---displaying definitions and example populations for categories, status and types, is shown in Figure 3.

The ORM metamodel for business requirements tracking as transformed to the ER attribute based model using IDEF1X as the implemented model is shown in Figure 4.

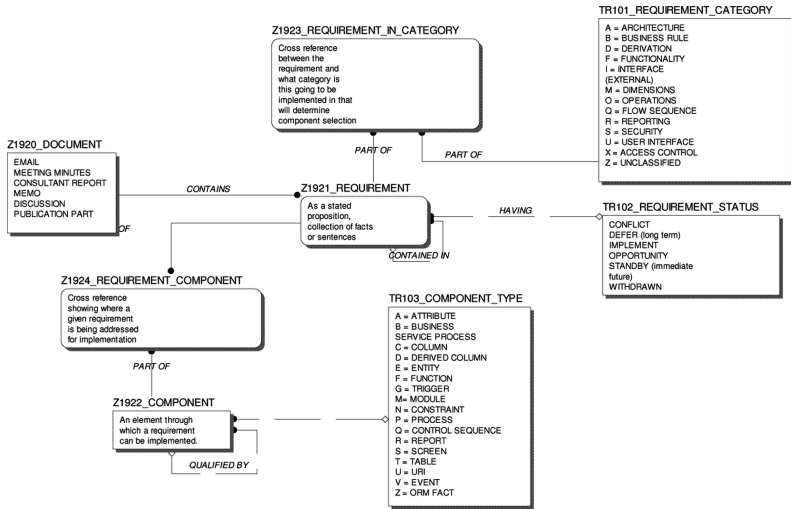


Fig. 3. ER based metamodel for Business Requirements transformed from ORM (Definitions)

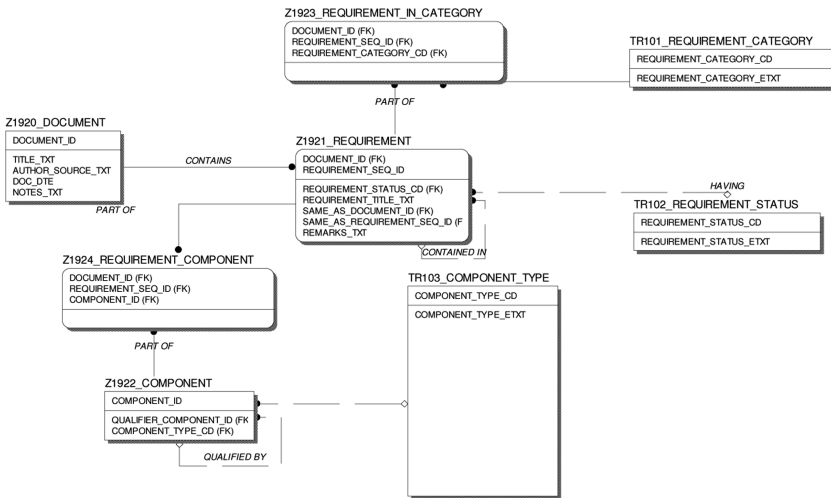


Fig. 4. ER based metamodel for Business Requirements transformed from ORM (as implemented)

Table 1. Example extract from on-going requirements tracking

DOCUME NT_ID	REQUIR EMENT_ SEQ_ID	REQUIRE MENT_ST ATUS_CD	REQUIREMENT_TITLE_TXT
IIS-004	40		Allow searching by manager name
IIS-004	41		Allow searching by company name
IIS-004	42		Allow searching by province
IIS-004	43		Capture contact's e-mail address
IIS-004	44		Produce a graph of companies having high viola- tions
IIS-004	45		Produce a graph of companies having high instance of certain violations
IIS-005R1	1		Allow user to view and edit tombstone data
IIS-005R1	2		Capture business type
IIS-005R1	3		Capture dangerous good handled
IIS-005R1	4		Capture means of containment used
IIS-005R1	5		Capture means of transport used
IIS-005R1	6		Keep a historical record of previous data values

It is important to note that for the trackability of the mappings across the multitude of models and implementation components, each model artefact and implementation component artefact is supported by a standardized form of identifier. Examples include Business Requirement identifier, Process identifier, process information flow identifier, fact type identifier, fact based constraint identifier, table identifier, column identifier, SQL constraint identifier, etc.

In its currently implemented form, the on-going collection of business requirements consists of reviewing over 150 documents collected over a span of 10 years. The makeup of the document set consists of previously defined business requirements documented in consultant reports, emails, memos, forms, graphs as requirements, project documentation, and reference materials like Acts, Regulations, guidelines that need to be conformed to. So far, we have extracted over 500 business requirements which are trackable based on the requirements traceability metamodel as implemented in Oracle 10g, a sample of which is shown in Table 1.

4 Corporate SDLC Correlation Mapping

Transport Canada uses Macroscopic Productivity Centre [5] as the SDLC standard set of documentation and practices in support of their IT system development projects. Each life cycle phase is supported by one or more sets of deliverable documents that contain specifications at major life cycle stages. The above ORM model as shown in Figure 2 already contains a construct identified with the fact type 'Document contains Requirement'. A Macroscopic document identifier is created and is associated with related requirements. In this way, the business requirements from various sources are captured and stated as propositions, and these propositions are captured and documented in the format and style as required by the stated template for Macroscopic documentation, thus meeting Transport Canada corporate standard requirements.

5 Conclusion

The business requirements traceability metamodel provides the much required requirements lineage and more importantly, captures all facets and incarnations of business requirements ‘as-they-happen’. The metamodel enables the tracking of documents, tracking of actual requirements, projected into the realization and implementation of the stated requirements. The metamodel allows navigability across multiple models involved in the systems development life cycle for an organization, and supports the zig-zag or other development processes like agile, waterfall, prototyping, spiral etc.

References

1. Mayer, R.: IDEF0 Functional Modeling. Knowledge Based Systems, Inc. College Station, TX (1990)
2. Object Management Group (OMG): Business Process Modeling Notation (BPMN) Specification, Final Adopted Specification (2006)
3. Mayer, R., Menzel, C., Painter, M., et al.: Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report, Knowledge Based Systems Inc., KBSI (1995)
4. Appleton Company, Inc.: Integrated Information Support System: Information Modeling Manual, IDEF1 – Extended (IDEF1X), ICAM Project Priority 6201, Subcontract #F33615-80-C-5155, Wright-Patterson Air Force Base, Ohio (1985)
5. Macroscopic ProductivityCentre, by Fujitsu Consulting, USA,
<http://www.fujitsu.com/us/services/consulting/method/macroscope/prodcentre/index.html>

Requirements Specification Using Fact-Oriented Modeling: A Case Study and Generalization

Gabor Melli¹ and Jerre McQuinn²

¹ Prediction Works, Inc.
gmelli@predictionworks.com

² Microsoft Corporation
jerrem@microsoft.com

Abstract. We present a case study of the application of fact-oriented modeling to the capture and management of requirement specifications for the introduction of an information technology solution within Microsoft. The delivered solution involves automation and centralization of information about relationships between Microsoft product offerings. The methodology contributed to the project's fast turn-around time and high quality deliverable largely due to the clarity, completeness and traceability of business concepts and individual specification statements. We conclude with a generalization of the methodology used.

Keywords: Fact-oriented modeling, Case Study.

1 Introduction

Fact-oriented Modeling¹ [3], [4], [7] is a technique that assists with the conceptual modeling of an IT Solution. The approach however has not yet been fully incorporated into software requirement specification standards [8], [9], [10], [12], [13], [14], [2]. With the introduction of such standards as Semantics of Business Vocabulary and Business Rules (SBVR) [5], [7] it is now easier to consistently employ Fact-oriented Modeling in the delivery of enterprise solutions. In this paper we intend to illustrate, via a case study, how a Solution Requirements Specification Methodology can integrate Fact-oriented Modeling with a classic requirements specification approach.

Fact-oriented Modeling depends upon a controlled vocabulary of Business Concepts which can be used by business and IT stakeholders to communicate in a common language, leaving little room for ambiguity. Many Microsoft legacy systems have physical data structures that do not reflect the business concepts and relationships that they support. Fact-oriented Modeling changes this paradigm by requiring that Business Concepts and the allowed actions and relationships between them are specified as Business Rules *before* the functional specification begins.

¹ Key terms used in this paper are defined in the factmodels.org/SRS1/v080630 and www.factmodels.org/PReM1/v080630 repositories

In Microsoft IT, we have developed a Structured Requirements Management (STREAM) [1] approach to documenting Specification Items by employing Fact-oriented Modeling and explicit identification of Business Concepts, Business Facts, and Business Rules in RuleSpeak® [6] notation. The authors, recently employed Fact-oriented Modeling to the specification of a new IT application, named PReM, to centrally manage product relationships. We represented the solution's Business Requirements by means of Specification Items that are atomic, itemized, prioritized, and written in a Structured Language that is understandable to both Business Stakeholders and IT Stakeholders.

We employed a binary fact-oriented approach and made closed-world assumptions because they were sufficient for our needs, and because we wanted to experience the benefits of adopting the simplest workable approach.

Perceived benefits from the use of the approach included:

- Specification Statement Clarity
 - Reduced guesswork in picking business concepts, facts, rules and requirements from paragraphs of text.
 - Encouraged the graphical depiction of the relationship of Business Concepts.
 - Increased the accuracy and speed of ensuing analyses (functional, technical, and test case analyses).
 - Promoted faster production of functional and technical specifications, even when the project was outsourced and off-shored. [15].
- Specification Statement Traceability
 - Enabled faster and more accurate determination of coverage by functional and technical specifications and test cases.
 - Sped up issue resolution because facts were interconnected with the logical data model and the physical data model.

In section 2 of this paper, we present PReM as an illustrative case study of a successful integration of Fact-oriented Modeling into requirements specification. In section 3, we then generalize, using Fact-oriented Modeling to illustrate the Solution Requirements Specification Methodology (SRS) process itself. Sections 4 and 5 then describe the benefits and make concluding remarks.

2 Case Study

This section presents a case study of the Product Relationship Management (PReM) application development, begun in the fall of 2007, to provide relationships between Microsoft products in computer-consumable format. Prior to this project, the relationships were either solely in legal documents or were maintained in an assortment of consuming systems.

The project used a classic waterfall Software Development Lifecycle Methodology (SDLC), described below, but the collection of *Specification Statements* would have equally well enabled an agile Scrum Methodology.

2.1 Vision-Scope Phase

The PReM Vision-Scope phase documented the business opportunity, constraints, solution alternatives, costs and benefits. It identified the scope of a recommended solution and a roadmap to realize the solution over multiple releases.

PReM set out to address the opportunity to better manage interrelationships between Microsoft products to help customers make optimal purchase decisions. For example, customers need to know that if they have Office Standard Edition L&SA they may “step up” to Office Professional Plus (Figure 1). Currently these product relationships are documented in policy documents, such as the Product Use Rights². The PReM vision is to systematically deliver a centralized master service for all Product Relationships that impact customer decisions.

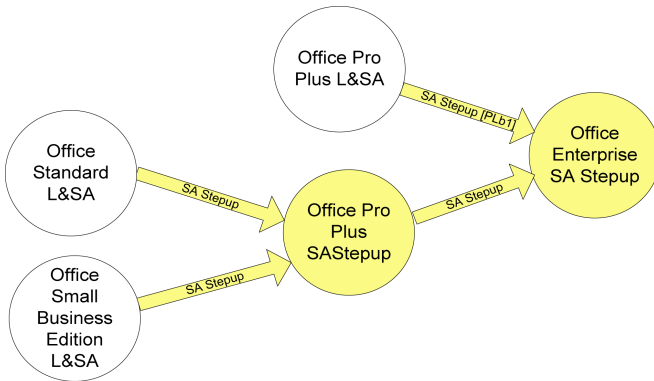


Fig. 1. Office SA Step-Up Relationships

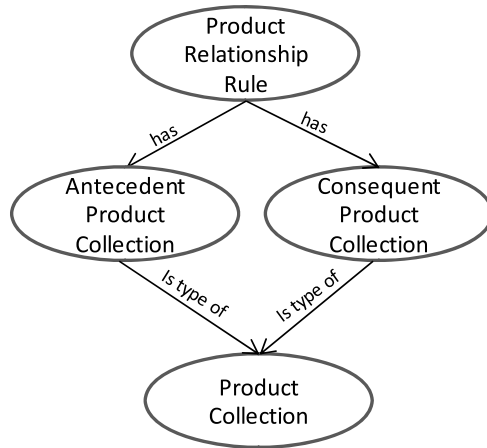


Fig. 2. Fact Model Snippet

² <http://www.microsoftvolumelicensing.com/userights/DocumentSearch.aspx?Mode=3&DocumentTypeId=1>

We began with Information Analysis of policy documents, of master product data, and of experts' knowledge. We quickly realized that the relationship instances were too numerous to be accurately created and maintained manually. In a typical case, each of 150 product part numbers (SKUs) relates to each of 150 different SKUs to generate 22,500 individual relationship combinations. There are hundreds of such cases and so we concluded the need to automate the relationship management using descriptive attribution for each SKU.

Further analysis gave insight to abstractions that were not rendered in the initial fact model. For example, we immediately saw that we needed the concept of Product Collection and Product Relationship Rule and that a Product Collection can be an Antecedent Product Collection or a Consequent Product Collection in a Product Relationship Rule. A snippet of the fact model is shown in Figure 2.

Since the volume of relationships requires automation, and because relationships undergo changes with the on-going launch and retirement of products, we specified a system which implements Business Rules of the following canonical form:

If a customer has/wants to install/order an offering from product collection C then, according to relationship type T, the customer is required/entitled/recommended to install/order/have acquired/align to offering(s) from product collections (A₁ AND A₂ AND ...) OR ... OR (...AND A_{n-1} AND A_n)

We refer to collection A as the as the “antecedent collection” and collection C as the “consequent product collection.” Below is a sample rule instance:

If a customer wants to order an offering ”Office Professional Plus SA Step-Up” then, according to relationship type “SA Stepup”, the customer is required to have acquired an offering from product collection “Office Standard” or “Office Small Business Edition”.

In the Vision-Scope phase we began to capture Business Concepts and Business Facts. It was during the Requirements Phase that we made a methodical and diligent commitment to expressing the solution as Specification Items.

2.2 Requirements Phase

The requirements phase completes the Information Analysis, and performs Use Case Analysis and Business Process Analysis to produce a Business Requirements Document (BRD).

Use Case Analysis followed from a fact model (Figure 3) which relates actors to Business Concepts discovered during Information Analysis. This analysis discovered new concepts and facts and Business Rules, expressed as RuleSpeak®.

Each role in the Use Case Analysis was described in a fact-oriented manner. Users found it easier to understand this fact model when we used person-icons for role concepts. The description of roles, such as the Rules Analyst, was critical to evangelizing the adoption of new responsibilities in operational teams.

We performed Business Process Analysis to specify the management of the new Product Collections and Product Relationship Rules. For instance, the “Define Relationship” process included steps for a Rules Analyst to receive a request, encode a Product Collection, encode a Product Relationship Rule, and commit them to production. Through Business Process Analysis we discovered more concepts, facts, rules and requirements.

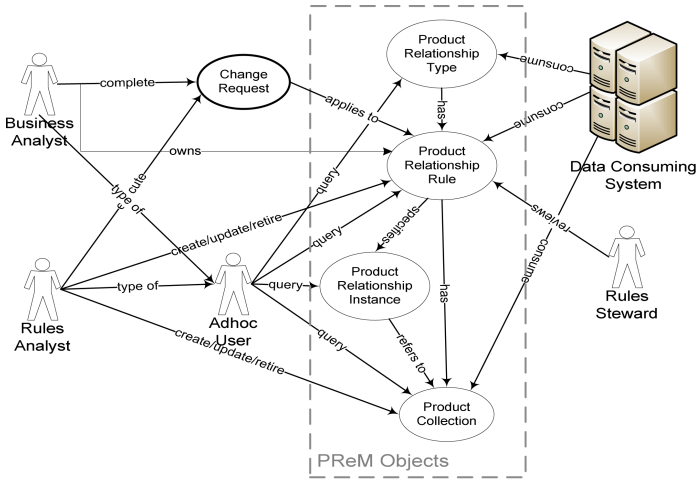


Fig. 3. Use-based Fact Model

ID	Process	Step	Type	Priority	Requirement Definition
108	Define Relationship	1.1	Rule	1	A Product Collection must be either an Attribute-based Collection or a Composite Collection.
123	Define Relationship	1.2	Fact	1	A Product Collection can act in the role of a Consequent Collection.
124	Define Relationship	1.2	Fact	1	A Product Collection can act in the role of an Antecedent Collection.
66	Define Relationship	1.1	Reqd	1	Solution shall provide an interactive Attribute-based Collection creation capability for Rules Analyst to choose attribute, and then select a list of values available for that attribute.

Fig. 4. Sample of the specification items master list within the BRD

Finally, we compiled all of the facts, rules and requirements into an Excel spreadsheet and gave each item a numeric identifier, alignment with process step, identification of type (Fact, Rule, Requirement), and a priority. This table became the master of the information (Figure 4).

The management of the spreadsheet motivated one of the authors (Melli) to develop an Access database. Requirements materialized for new capabilities of the Access database which in turn led to the application of Fact-oriented Modeling to a generalized Solution Requirements Specification Methodology, described in section 3.

2.3 Design and Build Phases

When the requirements phase completed, the direct involvement of the authors turned from a primary role to a supporting role. The project was designed and built by an

offshore-outsourced vendor solicited by means of a Request for Proposals (RFP) package. [15] Bidders were supplied with the BRD and it appeared that the fact modeling technique improved the quality of the proposals we received.

The design phase of the project resulted in a Functional Specification Document that included a Logical Data Model (LDM) and a Technical Specification Document that included a Physical Data Model (PDM). The Business Concepts translated directly into entities in the LDM and the PDM. (Figure 5)

The build phase results in the development and system test of working code. The developers referenced the fact model in this phase.

2.4 User Testing Phase

An independent team wrote User Acceptance Test (UAT) cases. Fact models assisted UAT authors and testers in rapidly grasping the relationship of Business Concepts to each other. UAT was executed in a shorter time, with significantly fewer bugs than projects of similar size and complexity. We maintained test cases in Microsoft Visual Studio with Specification Traceability from the BRD. (Figure 5)

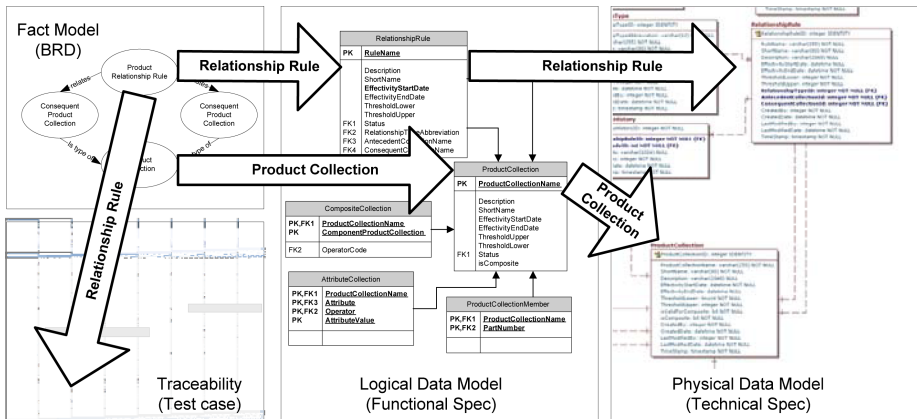


Fig. 5. Fact-oriented Modeling enabled Business Concepts to persist through system analysis, design and test cases without guesswork and without loss in translation

3 Methodology

This section presents a generalization of the solution requirements specification methodology (which we refer to as SRS) that was employed in implementation of the PReM project described in the use case. The aim of this section is to help the reader apply some or all of the methodology to their needs.

To illustrate how the methodology could be applied to another endeavor we present the SRS methodology using a structure similar to the Business Requirements Document

structure for the PReM project. A solution is described along two main subject areas: how agents interact with the business concepts (Business Process Analysis) and how business concepts relate to each other (Information Analysis). Each subject area includes a brief descriptive narrative, a fact model, and a description of the concepts included in the fact model.

3.1 Business Process Analysis

This section describes the agents in fact modeling, and the tasks they perform. This approach is consistent with the Business Analysis Body of Knowledge with one or more Business Analysts acting in different roles. [10] A Business Process Analyst produces “as-is” and “to-be” process flowcharts to identify process improvements to be automated. From Business Process Analysis they discover Business Requirement which may be further described via Use Cases.

Sometimes Business Process Analysis is augmented by Information Analysis. A Business Information Analyst might be specialized at performing database queries or experienced at policy interpretation.

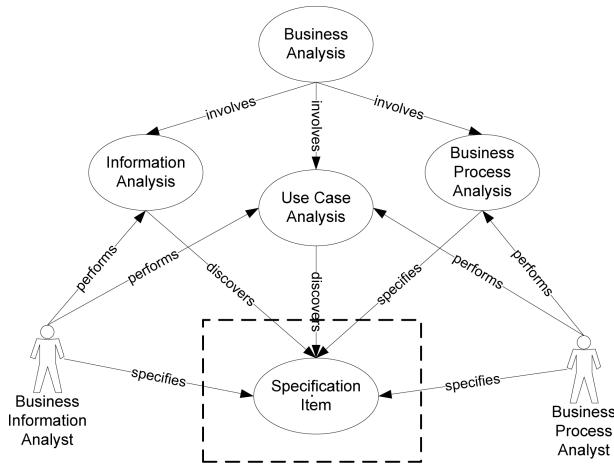


Fig. 6. A fact model focused on the users of the methodology and what they can affect. The node enclosed in the rectangle is described in section 3.3.

3.2 Information Analysis

This section describes the Information Analysis we performed to document the SRS methodology. We include a fact model (Figure 7) of the information objects and we developed definitions for the concepts in the model³. Although for PReM we did not identify different kinds of requirements, the figure shows how requirements might be classified into Business Requirements, User Requirements, etc. Classification of requirements varies somewhat between standards [10], [12]. [14].

³ The master repository for the methodology described in this paper can be found at http://www.factmodels.org/SRS1/v080630/SrsMgr1_SrsDb.accdb.zip

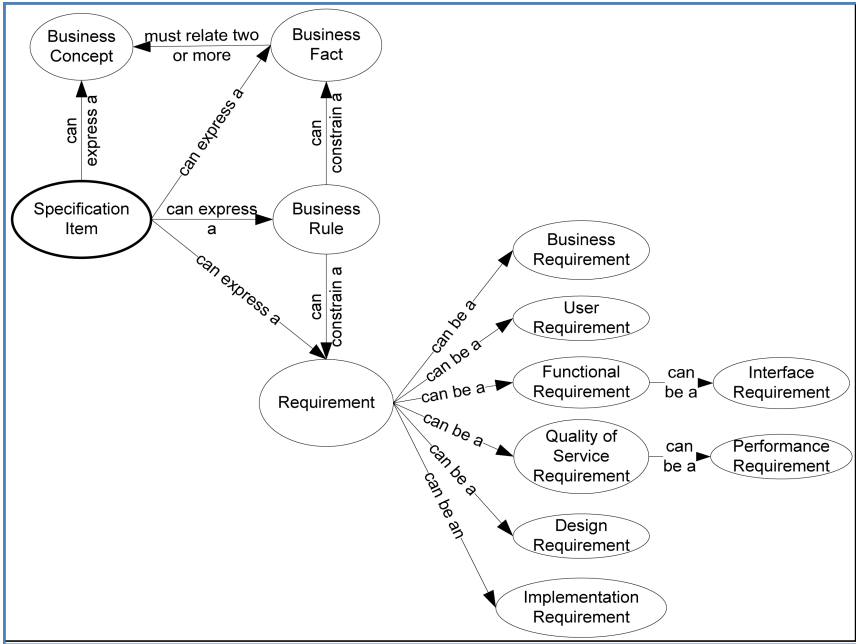


Fig. 7. Fact Model of the SRS Methodology

3.3 Specification Item

The central concept of the methodology is a Specification Item which provides an atomic, itemized and prioritized aspect of a Solution that is expressed in a Structured Language and is under the jurisdiction of the Business. We present it as a fact model in Figure 8, followed by the definition of Specification Item.

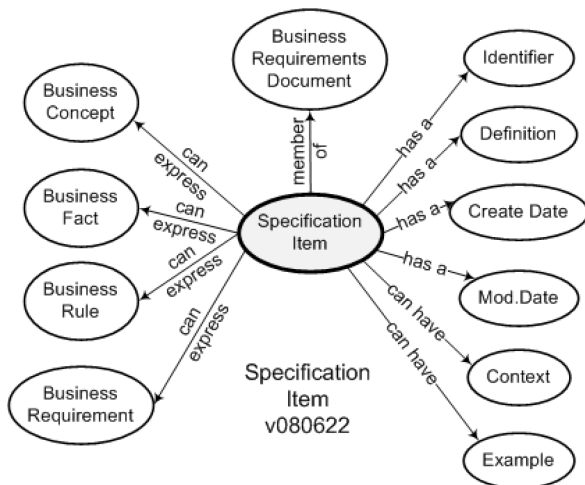


Fig. 8. Fact Model of a Specification Item

<i>Name</i>	<i>Definition</i>
Specification Item	<p>An atomic, itemized and prioritized aspect of a Solution that is expressed in a Structured Language and is under the jurisdiction of the Business.</p> <p>Fact(s):</p> <ul style="list-style-type: none"> • Can express a Requirement. • Can have a Subject Area. • Can have an Identifier. • Can have a Priority. • Can express a Business Concept. • Can express a Business Rule. • Can express a Business Fact. • Can have a Specification Statement. • Can have a Context Statement. • A UAT Test Case can validate a Specification Item.

Generalizing the methodology caused us to reflect on the specification process and helped bring clarity to the objectives of business requirements specification.

4 Benefits Analysis

4.1 Summary of Objects

This section presents a quantitative analysis of the number of information objects and relations that were generated as a result of the requirement specifications. While we have limited comparative metrics, the table below shows a comparison with three projects of similar size and complexity, managed by one of the authors (McQuinn) [11].

The comparison projects employed less-rigorous (Project 1) or no fact modeling (Projects 2 and 3). All experienced greater change, measured by BRD iterations and requirements change after approval. It is the authors’ feeling that greater rigor contributed substantially to better quality and faster delivery. This is corroborated by benefits listed in the following section.

	Item	PReM	Comparison Project 1	Comparison Project 2	Comparison Project 3
Basic metrics	Concepts	33	22	unspecified	unspecified
	Facts	40	23	unspecified	unspecified
	Rules	36	29	10	25
	Requirements	70	114	25	50
	Process Steps	18	18	26	13
	BRD pages	78	66	13	36
	Months from BRD complete to test complete	5	8	5	6
Indicators of rigor	BRD revisions to complete	6	13	unknown	56
	Percent Requirements change after BRD approved	2%	17%	high	high
	Development efficiency, measured as total rules and requirements divided by months (higher is better)	21	18	7	13

4.2 Benefits

Anecdotal benefits were collected from the BRD consumers, including the systems analysts, designers, test engineers, and from the authors' own observations. We were finishing this paper as the PReM system went into production, and we continued finding benefits that resulted from a clear and well-organized BRD. Nearly all users pointed to the fact model diagrams as helpful in their initial understanding of the requirements.

- Fact modeling kept us focused on Business Concepts, reducing the temptation to dive into logical or physical modeling in the requirements phase.
- Fact modeling provided a framework and rigor to identify and focus on Business Rules in context, rather than adding them after the fact.
- The BRD was included in the vendor solicitation RFP. The selected vendor stated that the fact models helped in more quickly preparing a more accurate proposal.
- Systems analysts remarked on the readability and clarity of the BRD. The PReM analyst produced only 24 formal questions compared to more than 100 for other BRDs of similar size and complexity.
- The BRD and functional specification were each completed in 1 month, compared to typically 6 or more weeks each. The functional and technical specification proceeded in parallel, partly because of strong fact models to guide both.
- The India-based vendor reported that throughout the design and build phase they continually referred to the Specification Items by number, both increasing accuracy and decreasing turnaround on questions, exacerbated by a 12.5 hour time difference.
- Fact modeling standardized names for critical Business Concepts and those names persisted through systems analysis, design, test, and user documentation. The consistency aided in communication and searchability of all documentation.

4.3 Barriers to Adoption

Perceived barriers to adopting the approach include:

- The effort required to integrate the approach into an organization's existing Software Development Lifecycle Methodology (SDLC) may require modification of the format or usual content expected in the requirements documentation.
- It may be more difficult for the business stakeholders to grasp the big picture. They may miss the business narrative if it is not presented in story format.
- There may be some initial resistance by the business to the development and use of a controlled vocabulary for Business Concepts and precise and sufficiently abstract Business Rules.
- Fact modeling does not seem to come easily to requirements authors who have been oriented to process flows and use case documentation. Explicit training and peer reviews are necessary for the adoption of fact modeling.

5 Conclusion

Fact-based modeling was successfully applied to the delivery of a new IT solution at Microsoft. The approach followed sped-up all activities during all project phases and resulted in a higher quality (more accurate and consistent) solution.

Looking ahead we foresee long-term opportunity in:

1. Evangelizing within Microsoft IT in order to deliver future versions of the solution in response to changing business requirements.
2. Improving the accuracy and speed with which a Rules Analyst can speak with a Business Analyst to elicit Business Rules.
3. Establishing a repository of fact model patterns to enable re-use. Topic areas to assist enterprise solutions would include: Product, Customer, Pricing, Contracts, and Governance.
4. Aligning fact modeling with other industry standards (UML diagramming) and to elicit non-rule-based requirements.
5. Automating the capture and management of specification items, including automatic assignment of identifiers; validation that every rule has a fact; validation that every concept is defined, and automated addition of hyperlinks from an MS-Word-based document to the requirements specification repository.

References

1. Choteborsky, P., Gerrits, R.: Business Rules Management without a Rule Engine: Does it make sense? In: Proceedings of 10th International Business Rule Forum Conference (2007)
2. Goedertier, S., Mues, C., Vanthienen, J.: Specifying Process-Aware Access Control Rules in SBVR. In: Paschke, A., Biletskiy, Y. (eds.) RuleML 2007. LNCS, vol. 4824, pp. 39–52. Springer, Heidelberg (2007)
3. Halpin, T.: Business rules and Object-Role modeling. *Database Programming and Design* 9(10), 66–72 (1996)
4. Halpin, T.: A Logical Analysis of Information Systems: Static aspects of the data-oriented perspective. PhD Thesis (1989)
5. Nijssen, S.: SBVR ~ Ground Facts and Fact Types in First-Order Logic. *Business Rules Journal* 9(1) (January 2008)
6. Ross, R.G.: Principles of the Business Rule Approach. Addison-Wesley, Reading (2003)
7. Object Management Group: Semantics of Business Vocabulary and Business Rules (2007)
8. Vos, J.: Is There Fact Orientation Life Preceding Requirements? In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 688–698. Springer, Heidelberg (2007)
9. Wan-Kadir, W.M.N., Loucopoulos, P.: Relating evolving business rules to software design. *Journal of Systems Architecture* 50(7), 367–382 (2004)
10. International Institute of Business Analysis (IIBA): Business Analysis Body of Knowledge (BABOK ®) v1.6 (2006)
11. McQuinn, J.: Decision Tables-From Specification to Operation. Proceedings of 10th International Business Rule Forum Conference (2007)

12. IEEE: Recommended Practice for Software Requirements Specifications. IEEE Std 830-1998 (1998)
13. Software Engineering Institute: Capability Maturity Model Integration (CMMI) for Development, Version 1.2. Technical Report CMU/SEI-2006-TR-008 (2006)
14. Wiegers, K.E.: Software Requirements, 2nd edn. Microsoft Press (1999) ISBN 0-7356-0631-5
15. Iacovou, C.L., Nakatsu, R.: A risk profile of offshore-outsourced development projects. *Communications of the ACM* 51(6) (June 2008)

A Model for Data Quality Assessment

Baba Piprani¹ and Denise Ernst²

¹ SICOM Canada

² DSI Now, Canada

babap@attglobal.net, denise.mcconnell@rogers.com

Abstract. One of the major causes for the failure of information systems to deliver can be attributed to data quality. Gartner's figures and other similar studies show the failure rate hovering at a plateau of 50% for data warehouses since 2004. While the true cause of poor data quality can be attributed to a lack of supporting business processes, insufficient analysis techniques, along with protecting oneself with the introduction of data quality firewalls for incoming data, the question has to be raised as to whether a data quality assessment of the existing data would be worthwhile or plausible? This paper defines a data quality assessment model that enables a methodology to assess data quality and assign ratings using a score-card approach. A by-product of this model helps establish 'sluice gate' parameters to allow data to pass through data quality filters and data quality firewalls.

Keywords: Data Quality Assessment, Data Quality Firewall, Data Quality filter, Data lineage, Type Instance.

1 Introduction

Did you know that in September 1999 a metric mishap caused the crash landing of a Mars bound spacecraft where NASA lost a \$125 million Mars orbiter because 2 engineering teams, for a key spacecraft operation, used different units of measure which resulted in failure of data transfer due to the mismatch?[2]. Did you know that a referee in the World Cup Soccer 2006 match between Australia and Croatia handed a soccer player 3 yellow cards before the player was sent off? The rule is 2 yellows results in a player being sent off [1]. Did you know that data warehouse success measures, or more appropriately stated, "failure rates or limited acceptance rates" have been in the range of 92% (back in late 1990s) to greater than 50% for 2007 [3][6]----a dismal record indeed.

So what do we mean by "failure"? The meaning of the term "failure" has been amplified by the Standish Group [4] with the interpretation that the "success" of the project refers to the project being completed on time and on budget with all features and functions as initially specified; or, the project being "challenged" refers to the project is completed and operational but, over-budget, over the time estimate, and offers a subset of features and functions originally specified; and being "impaired" refers to the project being cancelled at some point during the development cycle.

According to the Standish Group's 2003 CHAOS report, 15% of the IT projects "failed" and another 51% were considered "challenged", while 82% of the IT projects experienced significant schedule slippage with only 52% of required features and functions being delivered. For 2004, results show that 29% of all projects succeeded i.e delivered on time, on budget, with required features and functions; 53% were "challenged"; and 18% failed i.e. cancelled prior to completion or delivered and never used. A staggering 66% of IT projects proved unsuccessful in some measure, whether they fail completely, exceed their allotted budget, aren't completed according to schedule or are rolled out with fewer features and functions than promised [5].

2 Root Cause of Failures

Quality appears missing in the meaning of success or failure of a IT project. Lack of data quality appears to be the major culprit in the "failure" of IT projects. The traditional project management triangle is represented by cost, scope and schedule in Figure 1 with quality injecting throughout the cycle but often enough there no associated project deliverable. This gap is unexpected yet understood so how do we assess and close the gap?

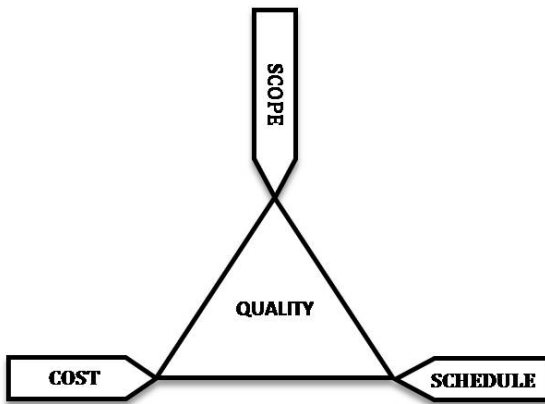


Fig. 1. Project Management triangle

It is important to observe that in any typical manufacturing process, quality is injected in every process from the very start. For example, a casting metal foundry technician systematically monitors a melt to assess that the required composition of metal compounds like carbon, iron, nickel, chromium, zinc etc. are in place prior to pouring to ensure the quality of the desired metal casting. Similarly, it is imperative that data quality needs to be injected in every phase of information system design and implementation with due diligence to governance, monitoring and auditing, among other things. In this paper we explore how we can define a similar quality control assessment for data.

3 Issues to Tackle

Where do we start? In our experience, examining the IT projects we have been called into to salvage and steer the usually sinking project towards 100% success, we have observed the following issues:

- Business requirements documentation is non-existent, not maintained upon change, too high-level, lacks integrated enterprise viewpoint, and lacks supporting business processes.
- Business rules are buried in program code which results in higher maintenance costs, dependency on specialized skills, and a lack of awareness.
- Undocumented definitions and missing semantics.
- Inability to audit and monitor changes to the architecture and contained data.

These are only samplings of the issues that are encountered that contribute to the data quality chasm in the building of information systems in both existing and under development. This list demonstrates a need to address data quality assessments throughout the solution’s Systems Development Lifecycle (SDLC) as in Figure 2.

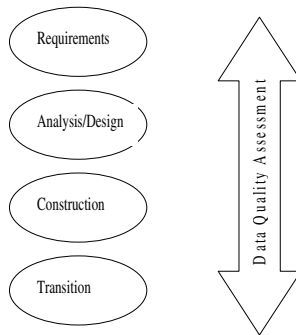


Fig. 2. Generic SDLC and data quality assessments

4 Data Quality Assessment Objective

The objective of the assessment is to identify the quality of the data in the identified business activity. An organization could be primarily in the service industry while another in the regulation sector.

The assessment results determine the accuracy, completeness, consistency, precision, reliability, temporal reliability, uniqueness and validity of the data. Assessment standard criteria are used when conducting the assessment.

When conducting an assessment, the business requirements are your window to the world. This can be a daunting task when assessing the quality of data at the enterprise perspective. Remember to scope the tests within the assessment criteria to ensure a balanced cost/benefit for the organization.

For example, if financial data is the vital to the business operations then the quality of such data will be an important factor in key business decision-making. Quality Assessment can happen in several manners, generally as either detection tests or penetration tests.

- Assessment detection tests assess data quality, identify risks, and can be used to determine risk mitigation efforts.
- Assessment penetrations tests, in addition to assessment detection, will penetrate systems with faulty data and monitor the effect and result. This will help to identify process deficiencies as well as determine quality of the data.

The summary of assessment tests should reveal data quality scorecard metrics vital to the organization's business and operations.

5 Methodology

Assessing data quality should not be like trying to pick up jello! Nor should it be an exercise in throwing darts on a Saturday afternoon in a pub! What is needed is an approach to methodically put in place data quality measures and standards sufficiently applicable at any stage of the life cycle, even if being parachuted in any part of the life cycle!

Then it becomes a primary requirement to be able to assess data quality across both the earlier stages and later stages of the development life cycle from any given point in the development life cycle. Not only that, it should be possible to precisely home in on any given stage of the development life cycle to enable the establishment of subsequent correctional measures going forwards.

Table 1 highlights how data quality assessment criteria are addressed by NIAM and ORM based modeling. The data quality assessment tests that can be conducted in level pair constructs across the 3 data lineage levels to determine the resultant data quality. The assessment test examples can be performed based on the data lineage level of the attribute e.g. by allocating each attribute in the implementation with a 'class-term' which simply groups similar attribute types based on similarity of concepts, e.g. amounts, dates, ratios, counts, quantities.

To achieve an organized approach for assessing data quality, the phases of a generic systems development life cycle could be aligned into 3 data lineage levels:

- Terminology and semantics
- "Type" - (metadata)
- "Instance" - (value).

The data lineage level pairs can be assessed in 'type – instance' level pairs. The 2 level pairs are "[terminology and semantics) + (Type metadata)]" and [(Type metadata) + (instance value)]".

The following criteria were found to be helpful in assessing data quality at each data lineage level as applicable (no ordering priority, alphabetical):

Table 1. Data Quality Assessment Criteria

Assessment Criteria	Description	Applying NIAM / ORM	Schema data assessment test
Accuracy	Degree of agreement between a set of data values and a corresponding set of correct values i.e. is the data correct	Natural language sentences with accompanying population diagrams and realistic sample data value populations related in a fact type that associated with a business "concept"	Where numeric data like amounts, counts, and quantities are involved, look for min and max ranges, less than or greater than zero or other figure etc. Where dates are involved look for future, past checks. Look for all of the percentages adding up to 100 or some defined limit.
Completeness	Degree to which values are present in the attributes that require them i.e. is the data complete	Null ability vs. totality as applied to a concept based fact type along with incomplete sentence populations in a population diagram.	Look for name required for regulated item records e.g. a NULL cannot own an aircraft If a large value property mortgage has been declared as being owned by a group, then the percent of the group ownership must add up to 100%
Consistency	Agreement or logical coherence in accordance with facts without variation or contradiction,	Using a "business concept" based focus; natural language sentences with sample real data values form the basis of agreement and understanding with the business. Expanding the natural language sentence constructs with variations and contradictions in population that break business rules.	Look for consistency in dates; like you cannot die in the future, you cannot be born in the future.
Precision	The quality or state of being exact, within the defined bounds and goals	The natural language elementary sentences and in particular, the elementary sentences form the basis of exact precision of the statement of the business fact in unambiguous terms	Look for precision in location latitude and longitude declarations; they must contain seconds in the Degree/Minute/Second system. Look for rounding errors
Reliability	Agreement or logical coherence that permits rational correlation in compari-	Using a "business concept" based focus, natural language sentences with sample real data values form the basis of agree-	Look for consistency of business types that an organization is licensed for and related types of returns or transactional consistencies

	son with other similar or like data	ment and logical coherence amongst the users to identify pattern similarities, e.g. risk measures used for different insurance companies based on insurance company groupings can be identified as being common across the company.	Look for lack of referential integrity on the use of same attributes being used in various tables
Temporal Relatability	Meanings and semantics that can change over time	Adding time dimension to the natural language sentences where the same fact type may undergo transformations.	Applicable to uniquely traceable items like serial numbers or particular licensed item identifiers, look for can the same item be involved with another item at the same time. Applies to ownership, involvement, and lineage. Look for loss of history data with no record of previous values
Timeliness	Data item or multiple items that are provided at the time required or specified	The granular level of the NIAM and ORM models provide easy cross correlation with data sets to be examined for timeliness in terms of availability and for the derivation of sequencing requirements thus directly affecting audit control.	Look for errors in versioning Data extractions Transactional details, extractions and loads to be monitored for same-period reliability Inability to relate data due to loss of history data
Uniqueness	Data values that are constrained to a set of distinct entries—each value being the only one of its kind	Every fact type must have at least one uniqueness constraint. This aspect is built into every sentence type and population diagram example for each concept and fact in NIAM and ORM	Look for dangling tables. Look for lack of referential integrity on the use of same attributes being used in various tables
Validity	Conformance of data values that are edited for acceptability—reducing the probability of error	The lower level granularity of a fact type associated with a business concept and the population diagrams helps define the acceptability and validity of data.	Look for artificial keys, identity values, system generated keys and apply at least one business key to a data grouping say in a data mart or row occurrence for a registry type data group (an inventory list like list of persons, list of vehicles etc)

5.1 Terminology and Semantics Data Quality

Taking advantage of the disciplined process for deriving business semantics and business rules in ORM, we see how a terminology based approach affects the basic data quality characteristics that can be used for assessing data quality.

In other words, use of SBVR with NIAM and ORM can contribute extensively to mitigating the risks in the approach to data quality assurance and definition of an enterprise model in the discovery of ‘missing business rules’.

5.2 “Type” (Metadata) Data Quality

Ideally, a well defined approach to assess data quality would be to first properly defining business requirements using say, SBVR using common terminology [8] and then following the bridging from the business model to an information system involving metadata and data models, and transforms for Terminology to Semantic Metadata and data models for:

- Business ontology to consolidated data and rules requirements
- Business requirements to class of platform independent data models
- Class of platform independent data models to class of platform specific data models
- Class of platform specific data models to vendor platform specific data models.

However, in reality it is a backstitched approach that works i.e. reverse engineer to look for ‘hidden business rules’ using some form of fact modeling methodology.

We see metadata here as being applicable to the result of a transform from terminology to data models---be they platform independent data models, platform specific data models or vendor platform specific data models.

The same above data quality characteristics assessment criteria can be applied to metadata at the level where a metadata registry-of-sorts e.g. ISO 11179:2003 Information technology — Metadata Registries (MDR) Part 3: Registry Metamodel and Basic Attributes [11]---is defined or available for an organization. Data quality assessment at the metadata level can now be performed with respect to other metadata elements in the registry in a similar fashion in correlation with other metadata elements---of course, with the similar transforms---being able to compare like metadata, particularly when different representations are involved with the same object type [9].

5.3 Instance (“Value”) Data Quality

One of the aspects that would be helpful in the assessment of the next level for “Instance” Data Quality (“value”) is the ability to match like data elements that form the metadata, i.e. a match between a “Customer_number” with “Customer_id”---which could provide different answers [9]. An important aspect in this regard is the use of standardized “class terms” for the data element names or categories. By identifying a data element e.g. an SQL column with meaningful data element names associated with corresponding constraints would also be part of the assessment, e.g. START_DTE, STOP_DTE; or AIRCRAFT_WEIGHT_NBR, PLATINUM_LOW_BALANCE_AMT etc. would also assist in establishing a data quality assessment criteria rules-set, where column suffixes _DTE, _NBR, _AMT are class terms applicable to SQL columns.

To correlate: A class term of column suffix `_DTE` could be associated with a requirement to be able to validate a range of dates, e.g. `STOP_DTE` must be greater than `START_DTE`. Additionally, a `STOP_DTE` could be defined as not being in the past and always must be greater than the `CURRENT_DATE` or be in the future. Similarly, the `START_DTE` must always be in the past or be the same as the `CURRENT_DATE` and never be in the future. Similarly, a `PLATINUM_LOW_BALANCE_AMT` for a platinum account cannot fall below \$1000 etc.

These are examples of how syntax based class terms for a column can be used to automatically derive constraining parameters, as imported and derived from the transforms of the SBVR NIAM or ORM based model.

6 Data Quality Metrics

The results of the Data Quality Assessment can provide operational metrics that can be used to continually monitor the quality of data. Where does one find metrics? An example of a simple metric could be the extent of implemented business rules vs. the total set of business rules (which can be determined via analysis using the Semantics and Terminology data lineage level).

In order to establish a metric of this nature, it is necessary to be able to derive the existing and missing business rules from a common template which should apply consistently no matter which level you are at--the instance (value) level, the type (metadata) level or the terminology or semantic level. In other words, you are looking for the same rule being interpreted across data lineage levels towards implementation.

An optimal method to achieve the set of "missing business rules" is to reverse engineer the semantics from the 'instance' level, or even at the type level. For example, given a set of values for aircraft mark identifiers (the visible letters identifying any aircraft), and a set of owners of the aircraft, we can look for things like: is it mandatory that an aircraft being registered needs to be owned by a party that has an address and contact, or can the same serial number engine exist on more than one aircraft at the same time, or can the same aircraft mark be on more than one aircraft at the same time etc. This question set can only be answered by conducting a quick reverse engineering or backstitching exercise from the attribute driven (ER, UML) relational or object based model to a fact based semantic modeling technique (ORM, NIAM, COGNIAM) to discover all hidden business rules. Some proponents might balk at the reverse engineering exercise, but if one examines the instance and type level paired combinations as part of a fact based sentence type, then it is a simple matter to validate that fact type!

The defined metrics can drive a dashboard-style data quality assessment scorecard. A scorecard could be in terms of reliability based on how the data has been preserved over the life cycle since inception, based on say, continuity or commitment to establish data currency by organizational personnel.

For example, a metric for particular data element, which, after going through the 3 data lineage levels in data quality assessment, could be assigned a value of "Yellow", while another data element could be assigned "Red"---meaning the "Yellow" tagged element is to be relied upon with caution (again, this can be further defined with a breakdown as to if it is coming from the Western Region operations it is 60% reliable

vs. Eastern Region operations with only 40% reliable, while the “Red” tagged element could simply state that this data element is not reliable.

Some of the additional activities [10] that support a scorecard include:

- Defining acceptable parameters and tolerances
- The metrics model that includes high risk situations based on data quality tolerances being met, establishing risk mitigating measures and providing definitive assurances vs. hand-waving or ‘I think...’ scenarios
- Go / no-go thresholds
- a metrics evaluation model
- process model to feed metric values into the scorecard.

What this is demonstrating is that based on an assessment methodology using Semantics, Metadata, and Instance, one is able to derive a dashboard equivalent value for a given metric that is usable in the scenario for reliability and dependability of the quality of data.

7 Author’s Experiences

In the author’s experience over the past 2 decades, assessing data quality has provided significant return on investments in varying private sector industries as well in the public sector.

The data quality assessment methodology as outlined in this paper has been used to recover seriously failing IT development / data warehousing projects; as a basis for Master Data Management; common object mappings across heterogeneous applications; in data warehousing; creation of metadata repositories; data quality firewalls; define enterprise data standards; establishing data quality scorecard; and the list goes on.

The approach has uncovered exceptional findings such as: Non-existing financial coding for expenses charged for over \$120 million; \$150 million recorded as amount already spent in the year 2097; Out of 26000 private owner records for a ‘regulated licensed item’, nearly 16000 were rejected due to inconsistencies like null address, lack of a proper or complete address, null name, names like Theodore and Alvin Chipmunk, Mickey Mouse etc.; In a mid-sized organization, there are over 100 staff members in an organization who have a birth date of 2061; 100% of HR personnel data was rejected when assessed against ~1000+ ‘business rules’; Entities reporting on the number of hours flown in a given year as 10000 hrs. (Note the maximum hours in a year is 365 days x 24 hrs = 8760 hrs).

A few major success stories include, all of which were on time and under budget: A large financial data warehouse with over 200 entities designed with 100% data quality at the schema level including auditing and monitoring tools; It took a remarkable 90 days with a 8 person team consisting of subject matter experts and technical staff.; The warehouse continues to be operational and has never once been re-loaded; Operational maintenance consists of 2 technical staff and 1 business analyst supported by the infrastructure-working group; Over 100 users accessing 3 cubes and 50+reports; Implementation of a previously failed re-design of a financial system used in the regulatory sector; Resurrected, assessed, data converted and operational within 2 months with a 5 person combined business/technical team; Data quality scorecard used by management.

References

1. BBC Sport: Ref Poll sent home from World Cup,
http://news.bbc.co.uk/sport2/hi/football/world_cup_2006/5108722.stm
2. CNN.Com: Metric mishap caused loss of NASA orbiter,
<http://www.cnn.com/TECH/space/9909/30/mars.metric.02/>
3. Gartner Group Report: Gartner Press Release, Gartner Website – Media relations (2005),
http://www.gartner.com/press_releases/pr2005.html
4. Standish Group International, Inc.: Chaos Chronicles and Standish Group Report (2003),
http://www.standishgroup.com/sample_research/index.php
5. Standish CHAOS Chronicles: Lessons From History,
<http://lessons-from-history.com/Level%202/Project%20Success%20or%20Failure.html>
6. ITtoolbox Blogs, Madsen, Mark: A 50% Data Warehouse Failure Rate is Nothing New,
<http://blogs.ittoolbox.com/eai/rationality/archives/a-50-data-warehouse-failure-rate-is-nothing-4669>
7. Zachman, J.A.: A Framework for Information Systems Architecture. IBM Systems Journal 26(3) (1987); IBM Publication G321-5298
8. Chapin, D., Hall, J., Nijssen, S., Piprani, B.: A Common Terminology, Semantic Metadata & Data Model Framework for Relating SBVR, ISO 704 & 1087 to ISO/IEC 19763 & 11179. In: Metadata Open Forum, Sydney (2008),
<http://metadataopenforum.org/index.php?id=34,132,0,0,1,0>
9. Piprani, B.: Using ORM in an Ontology Based Approach for a Common Mapping Across Heterogeneous Applications. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 647–656. Springer, Heidelberg (2007)
10. Piprani, B.: Using ORM Based Models as a Foundation for a Data Quality Firewall in an Advanced Generation Data Warehouse. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1148–1159. Springer, Heidelberg (2006)
11. ISO 11179: Information technology — Metadata Registries (MDR) Part 3: Registry Metamodel and Basic Attributes, International Standards Organization, Geneva (2003)

Verbalization for Business Rules and Two Flavors of Verbalization for Fact Examples

Maurice Nijssen and Inge Lemmens

PNA Group, Geerstraat 105,
6411 NP Heerlen,
The Netherlands
{maurice.nijssen, inge.lemmens}@pna-group.nl

Abstract. In the literature of NIAM, ORM, CogNIAM, OWL, Business Rules and SBVR [1, 5, 7, 16, 18] one increasingly encounters the modeling process of verbalization. Most fact based conceptual analysts are aware that process models need to be extended with fact schemas including concept definitions as well as concrete examples of input and output, satisfying the conceptual schema. Not adding this extension to process models regularly leads to misinterpretation and low productivity. Could there be a misunderstanding with respect to the process of verbalization as used in the various fact orientation approaches? In this paper we demonstrate that there are three *quite different* verbalization processes that have so far been referred to by the process name ‘verbalization’, resulting in quite different output. We will argue that all three types of verbalization are useful. To avoid further misunderstanding we propose to call these Verbalization for Business Rules and Verbalization for Fact Examples with and without using a fact type form (fact pattern), respectively, or more in the style of SBVR Structured English: (1) Verbalization with keywords, (2) Verbalization using a fact type form without keywords and (3) Verbalization without using a fact type form and without keywords. Each of these has a specific aim and each is useful in conceptual modeling.

1 Introduction

In this paper we will demonstrate that there are three useful types of verbalization processes. Such verbalization modeling processes are an essential part of domain-specific and generic (meta) conceptual schema modeling as well as business communication. The importance of processes for modeling has been stressed in the literature [3, 6, 8, 9, 11, 14, 15]. Although the process ‘verbalization’ has been used extensively in the last 35 years in the fact orientation community [11], it turns out now that there are three different types of verbalization. The first we will call Verbalization for Business Rules and the second and third Verbalization for Fact Examples, *with* and *without* using a fact type form. “Object-Role Modeling (ORM) is a fact-oriented approach for modeling, transforming and querying information in terms of the underlying facts of interest, where facts and rules may be verbalized in language readily understandable by non-technical users of the business domain” [2].

2 Is There a Problem with Respect to Verbalization in the ORM Community?

As we will be making extensive use of the CogNIAM knowledge triangle, the triangle is introduced in figure 1:

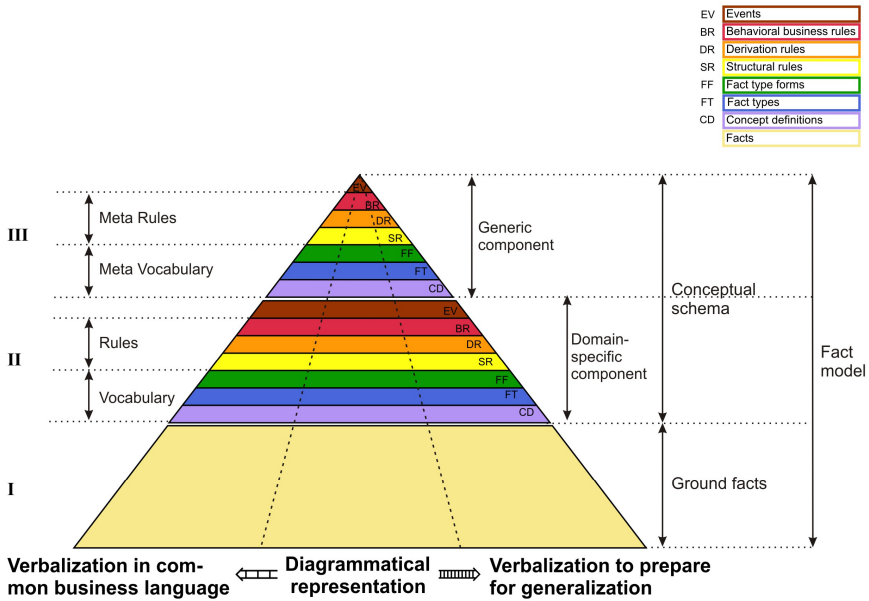


Fig. 1. Knowledge Triangle

In figure 1, ‘Verbalization in common business language’ should be interpreted as a synonym for ‘Verbalization for Business Rules’; ‘Verbalization to prepare for generalization’ is a synonym for ‘Verbalization for Fact Examples’.

In the ORM community the word ‘verbalization’ is used interchangeably for three very different purposes. Do we have a clear case of an undetected homonym? As this is hardly recognized, ambiguity and misunderstanding may occur.

Let us give an example from the SBVR specification, annex E.2.2.1.9 [13]:

<rental> incurs <late return charge>
 <rental> incurs <late penalty charge>

These are two specific fact type forms (each representing a fact type) at the domain-specific level (level II in the knowledge triangle (figure 1)). In the CogNIAM knowledge triangle these are specified in the middle vertical lane, the lane used by conceptual modelers (see figure 1). A rule for the associated fact types could be:

Each <rental> incurs at most one <late return charge>.

This rule is included in the left part of the middle level (level II), aimed at business persons that prefer to read rules in unambiguous Structured English. The step is going from the middle part of the middle level (level II) to the left part of the middle level (level II). On page 11 of Halpin and Morgan's book [9] the word *verbalize* is used to denote this process.

On page 9 of the excellent book of Halpin and Morgan's book the word *verbalize* means taking the middle part of the lowest level (level I) as input, use the fact type form of the middle level (level II) and produce the natural language sentences of the right part of the lowest level (level I).

If we would use the above two fact type forms of E. 2.2.1.9, and apply the above described verbalization process, we would obtain sentences for the right part of the lowest level (level I) of the following format:

- (1) 1122334455 incurs 150
- (2) 1122334455 incurs 170
- (3) 1234567890 incurs 180

What is the interpretation of 150 in sentence 1? Is it a late return charge, or a late penalty charge? Hence there is ambiguity.

3 How to Solve This Ambiguity?

This ambiguity can be solved by distinguishing two kinds of verbalization and the associated two kinds of fact type forms. The two kinds of fact type forms are needed if we want to eliminate the ambiguity. Both have been used in ORM and SBVR interchangeably.

Hence for the verbalization with the aim to express a (business) rule in unambiguous Structured English (Verbalization for Business Rules) the two fact type forms given above suffice. However, for the Verbalization for Fact Examples using a fact type form, not being rules, *another* fact type form pattern is recommended. In this case it would be:

rental <rental> *incurs late return charge of* <late return charge> Euros
rental <rental> *incurs late penalty charge of* <late penalty charge> Euros

The words in bold type are the qualifications of the nearest role they are bound to. In case of ambiguity, the association is explicitly stored.

If we would use these last two fact type forms – let us call them *qualified* fact type forms – we would obtain the following three natural language sentences:

- (11) Rental 1122334455 incurs late return charge of 150 Euros.
- (12) Rental 1122334455 incurs late penalty charge of 170 Euros.
- (13) Rental 1234567890 incurs late penalty charge of 180 Euros.

This ambiguity problem is one of the problems that can be eliminated with the systematic use of the CogNIAM knowledge triangle and associated related concepts.

Please note that the initial step of verbalization where an end user talks to a colleague does *not* make use of a fact type form, as it is not yet available in the development phase at the time this kind of verbalization is used. However, when the fact type form is derived in the Conceptual Schema Design Procedure [9], this fact type form can be used in the communication of ground facts. This Verbalization for Fact Examples using a fact type form (qualified) is illustrated in Figure 2.

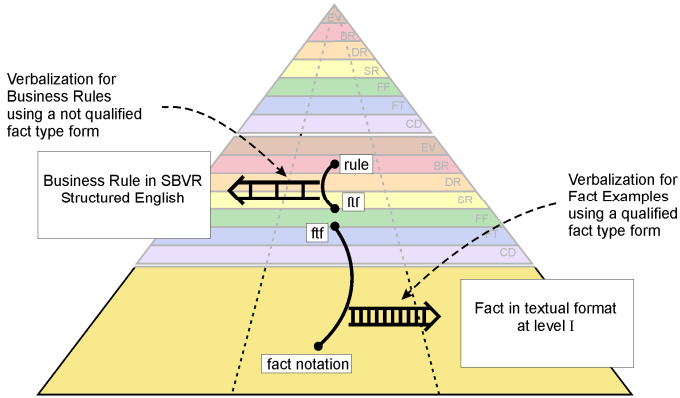


Fig. 2. Two different kinds of verbalization

4 Verbalization for Business Rules (Using a Fact Type Form)

For the process “Verbalization for Business Rules”, the verbalizer takes the fact type form (not qualified) and the rule expressed in diagrammatic format and adds one or more keywords to get a rule in SBVR Structured English. Example:

Fact type form:

rental *incurs* late return charge

Verbalization towards rule:

Each rental *incurs* at most one late return charge.

Above form is nearly always used in SBVR.

5 Verbalization for Fact Examples Using a Fact Type Form

For the process “Verbalization for Fact Examples (using a fact type form)” the verbalizer takes the given representation of facts (at level I), uses the fact type forms (of level II) and produces fact examples at the ground fact level (level I). Example:

Fact type form:

rental <rental> *incurs* late return charge of <late return charge> Euros

Values to be used:

1122334455; 150

Verbalization towards natural language sentences:

Rental 1122334455 incurs late return charge of 150 Euros.

In the paper Carver and Halpin recently presented at the EMMSAD conference [4], the authors asked the question with respect to normalization:

What went wrong? A research-historical, psychological excursus

When such a fairly obvious error in a standard, accepted theory goes undetected for three decades, one cannot help but ask what went wrong. [...] Thus, even more interesting than the question what went wrong originally, is the question why no one detected it for so long. [...] The idea that their assumption was flat wrong, was too radical a thought to occur to anyone. [...] A contributing factor to this oversight, however, seems to have been the aforementioned, and mathematicians' natural tendency to focus on the syntax – to the neglect of semantics, in this case.

Does the ORM conference community have the same tendency?

6 What Caused the Misunderstanding?

We could demonstrate that two kinds of verbalization produce the same result at the level of the knowledge triangle where there are only ground facts [13], i.e. facts that have no grammar function. However, if we apply the same types of verbalization at the domain-specific component of the conceptual schema or the generic component, then the results are *very different*.

7 Verbalization for Fact Examples without Using a Fact Type Form

In section 3 we discussed the Verbalization for Business Rules. Now we use the same fact type which is represented by the fact type form rental incurs late return charge for the Verbalization for Fact Examples without using a fact type form. In this case we do not have the fact type form, as it is not yet available as we are in the development phase of deriving this fact type form. Therefore we have to verbalize this fact type. In table 1 the Verbalization for Fact Examples without using a fact type form is given with respect to the role of the fact type. We have given the fact type the abbreviation rilrc.

Table 1. The result of verbalizing the fact type's roles

Fact type	rilrc	has role	rental	.
Fact type	rilrc	has role	late return charge	.

These two different kinds of verbalization at the level of the domain-specific component of the conceptual schema are shown in Figure 3.

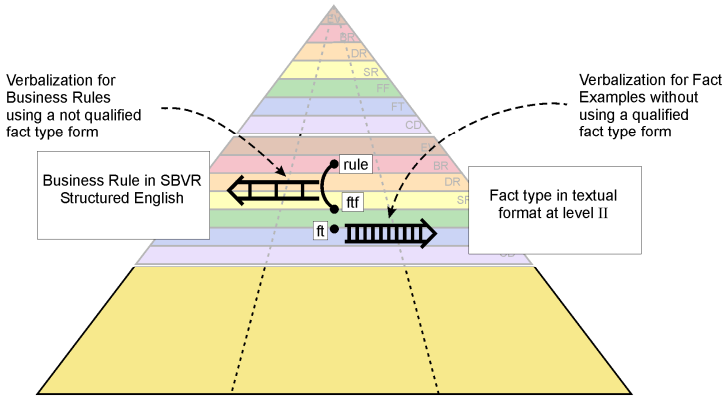


Fig. 3. Two different kinds of verbalization at level II

As shown above, by making systematic use of the knowledge triangle, these misunderstandings can be avoided.

The diagrammatic representation of the knowledge triangle of figure 1, 2 and 3 has been used in the last eight years by PNA as part of an approach that is now called CogNIAM. The knowledge triangle contains the core concepts of CogNIAM [12] in an easily memorizable way and in diagrammatical coherence. It can very quickly be seen that there are three levels, two of which share the same structure. These levels are:

- I. facts without a grammatical function, called ground facts in SBVR [13],
- II. facts with a domain-specific grammar function, called the domain-specific component of the conceptual schema in SBVR and
- III. facts with a generic or meta grammar function, called the generic component of the conceptual schema in SBVR.

The knowledge triangle has 7 knowledge classes in the domain-specific and generic component of the conceptual schema. One of these classes, events, is not part of SBVR. Why are events incorporated in the knowledge triangle? To facilitate respectful discussions with other communities such as UML.

The major common ground of ORM, SBVR, OWL, and CogNIAM is facts. The knowledge triangle uses nine parts (3 levels, at each level a left, middle and right side) to show the coherence between:

- A. A graphical or report representation of ground facts (middle part of the lowest level (level I)) and its textual representation (right and left part of lowest level).
- B. The textual or graphical representation of the domain-specific component of the conceptual schema (middle part of middle level (level II), having 7 knowledge classes) and the verbalization of rules in facts (right part of the middle level) as an *intermediate* step towards the next level, the generic level of the conceptual schema. The step from the middle part of the middle level to the left part of the middle level is extensively used in SBVR.

Please note that there is a distinction between three types of verbalization: one kind is used to express a rule in 'language that is readily understood by the business domain expert' [13, 10.1.1.2] and this is presented in annexes C, F and I of

SBVR. This is without doubt a useful form of verbalization. We call this Verbalization for Business Rules. However, CogNIAM has during many years used a second and third type of verbalization productively, to show the total coherence of all facts at all levels. One of the functions of Verbalization for Fact Examples without using a fact type form is to be a step towards the next level of grammar. This makes it possible to demonstrate that there are only three levels of facts and that the meta level describes itself, a major intellectual step in total understanding of the structure and modeling of conceptual schemas. The self description is clearly visible in figure 4, at the top of the knowledge triangle. The Verbalization for Fact Examples using a fact type form is used in the business communication of ground facts.

- C. The textual or graphical representation of the generic component of the conceptual schema (middle part of the upper level (level III)) and the verbalization of rules in facts as an intermediate step (right part of the upper level) towards the ‘next’ level. Because the generic component is self-describing, in this case the next level is the same (i.e. third) level. The step from the middle part of the upper level to the left part of the upper level is extensively used in SBVR.

By using two steps, Verbalization for Fact Examples without using a fact type form and generalization, from a best practice in requirements engineering, it is possible to demonstrate in a fairly easy to follow and understandable way that there are only three levels of facts and the coherence of major concepts of the Fact Orientation Approaches. SBVR is the first official specification or standard in business computing where concept definitions are first class citizens. The concept definitions form the bridge between the formal and the informal world, hence are vital for business communication. One of the 7 knowledge classes at the domain-specific and the generic level, Concept Definitions form the basis for each of the conceptual schemas, the domain-specific component and the generic component. The major concepts to be introduced are:

- Fact instances
- Concept definitions
- Fact types
- Fact type forms (with their subtypes: sentential forms and noun forms)
- Structural rules (Constraints)
- Structural rules (Derivation rules)
- Behavioral business rules and
- Event rules.

8 Summary and Recommendations

We started from the level of ground facts (level I in figure 4 above). In the old days one would say we started at the database level. At this level the facts have no grammatical function. By applying verbalization for fact examples (arrow 1) to the diagrammatic representation at the ground fact level, the results are the facts in a textual

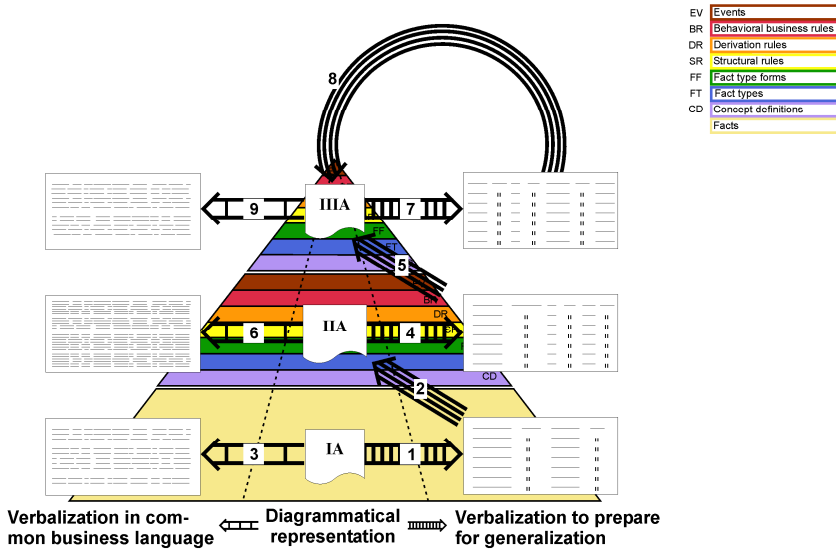


Fig. 4. Knowledge triangle with process aspects

format. By applying generalization (arrow 2) to the textual representation at the ground facts level the core of the domain-specific component of the conceptual schema was obtained in a diagrammatic format.

This diagrammatic format of the domain-specific component of the conceptual schema describes the meaning of terms at the ground fact level and it specifies the rules for fact populations, fact population transitions and for fact generation. Hence this determines the next lower level of facts and describes their semantics.

Next, by applying verbalization for fact examples (4) to the diagrammatic representation at the level of the domain-specific component of the conceptual schema, we obtain the textual format of the domain-specific component.

Continuing this process, by using generalization (5) at level II, the result is a diagrammatic representation of a core part of the generic component of the conceptual schema. This diagrammatic format of the generic component of the conceptual schema stipulates (6) the semantics and rules for the domain-specific component of the conceptual schema. Again, by applying Verbalization for Fact Examples (7) to the diagrammatic format of the generic component, we obtain a textual representation of a core part of the generic component of the conceptual schema.

As was illustrated previously, by applying generalization (8) at level III, the result was the identical representation of the generic conceptual schema, i.e. there is no higher conceptual level than level III.

The beauty of the generic component of the conceptual schema is that in effect it stipulates itself (9)!

It is our recommendation to distinguish the three kinds of verbalization by introducing clear terminology: we call the first kind of verbalization Verbalization for Business Rules (constraints, derivation rules or rules of guidance for humans) or Verbalization *with* Keywords. The second type we call Verbalization for Fact Examples

using a fact type form or Verbalization using a fact type form and *without* Keywords. The third we call Verbalization without using a fact type form and *without* Keywords. The major function of the process Verbalization with Keywords is to have the rules understood by business experts. The major function of Verbalization using a fact type form and without Keywords is to be able to communicate ground facts for business communication-purpose. The major function of Verbalization for Fact Examples without using a fact type form and without keywords is to have a deterministic procedure to derive the domain-specific component of the conceptual schema, to derive the generic component of the conceptual schema, often called the meta-schema and to demonstrate that the same procedure applied to the meta-schema will result in (a subset of) the meta-schema. Hence, all three types of verbalizations are useful, and each has its audience.

References

1. Anderson Healy, K.: Special Report on SBVR. *Business Rules Journal* 9(3) (2008) ISSN: 1538-6325
2. Balsters, H., Carver, A., Halpin, T., Morgan, T.: Modeling Dynamic Rules in ORM. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops*. LNCS, vol. 4278, pp. 1201–1210. Springer, Heidelberg (2006)
3. Bollen, P.: Using Fact-Oriented for Instructional Design. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops*. LNCS, vol. 4278, pp. 1231–1241. Springer, Heidelberg (2006)
4. Carver, A., Halpin, T.: Atomicity and Normalization. In: *Thirteenth International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD 2008)*, Montpellier, France (2008)
5. Chapin, D.: SBVR: What is now Possible and Why? *Business Rules Journal* 9(3) (2008), <http://www.BRCommunity.com/a2008/b407.html>
6. Damien, T., Vereecken, J., Christiaens, S., de Leenheer, P., Meersman, R.: T-Lex: A Role-Based Ontology Engineering Tool. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops*. LNCS, vol. 4278, pp. 1191–1200. Springer, Heidelberg (2006)
7. Hall, J.: Business Semantics of Business Rules. *Business Rules Journal* 5(3) (2004), <http://www.BRCommunity.com/a2004/b182.html>
8. Halpin, T., Curland, M.: Automated Verbalization for ORM2. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops*. LNCS, vol. 4278, pp. 1181–1190. Springer, Heidelberg (2006)
9. Halpin, T., Morgan, T.: *Information Modeling and Relational Databases*. Morgan Kaufmann, San Francisco (2008)
10. Hansen, J., dela Cruz, N.: Evolution of a Dynamic Multidimensional Denormalization Meta Model Using Object Role Modeling. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops*. LNCS, vol. 4278, pp. 1160–1169. Springer, Heidelberg (2006)
11. Nijssen, S.: On Experience with Large-scale Teaching and Use of Fact-Based Conceptual Schemas in Industry and University. In: *Proceedings of the IFIP WG 2.6 Conference on Data Semantics*, North-Holland Publishing Company, Hasselt (1986)
12. Nijssen, S., Bijlsma, R.: A Conceptual Structure of Knowledge as a Basis for Instructional Designs. In: *Proceedings of the Sixth International Conference on Advanced Learning Technologies (ICALT 2006)*, pp. 7–9. IEEE Computer Society, Los Alamitos (2006)

13. OMG (Object Management Group), Semantics of Business Vocabulary and Business Rules (SBVR), v1.0. Online as document 08-01-02 (2008),
<http://www.omg.org/spec/SBVR/1.0/PDF>. SBVR 1.0 and supporting files
<http://www.omg.org/spec/SBVR/1.0/>
14. Piprani, B.: Using ORM-Based Models as a Foundation for a Data Quality Firewall in an Advanced Generation Data Warehouse. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1148–1159. Springer, Heidelberg (2006)
15. Ross, R.: Business Rule Concepts: Getting to the Point of Knowledge, 2nd edn. Business Rule Solutions LLC, Houston (2005)
16. Ross, R.: The Emergence of SBVR and the True Meaning of 'Semantics': Why You Should Care (a Lot!) ~ Part 1. Business Rules Journal 9(3) (2008),
<http://www.BRCommunity.com/a2008/b401.html>
17. Sowa, J.: Fads and Fallacies about Logic. IEEE Intelligent Systems 22(2), 84–87 (2007),
<http://www.jfsowa.com/pubs/fflogic.htm>
18. Vanthienen, J.: SBVR: The ABCs of Accurate Business Communication. Business Rules Journal 9(3) (2008), <http://www.BRCommunity.com/a2008/403.html>

How to Avoid Redundant Object-References

Andy Carver

Neumont University, Utah, USA
andy.carver@student.neumont.edu

Abstract. A type of data “redundancy” that is not fact-redundancy arises from object-references that are non-“information-bearing”. Several forms of this phenomenon may occur in fact samples, including mention of a scope-defining object, and use of an anaphoric term – whether stated or implicit. We give various examples of this phenomenon of non-informative object-reference, and suggest that the problem is addressed by fully-semantically-accurate modeling: if we can correctly capture all referential meaning (which requires working with fact instances, not just fact types) – including whether the object-reference is intended as “information-bearing” (in its context) – then a design-procedure exists, outlined here, that will attribute fact-type “roles” to mentioned objects in such a way as to avoid all non-information-bearing object-reference.

1 Introduction: The Problem of Redundant Object-Reference

The sort of data redundancy we are calling “redundant object-reference” was partially addressed in a paper which we co-authored for the 2007 ORM Workshop [3]. The focus of that paper, however, was not on what caused the redundant data structures, but rather on schema-transformations that would repair certain cases of such redundancy. The redundancy itself was taken for granted, not analyzed as to its nature or how it might have been avoided in the first place. One example given there we repeat here:

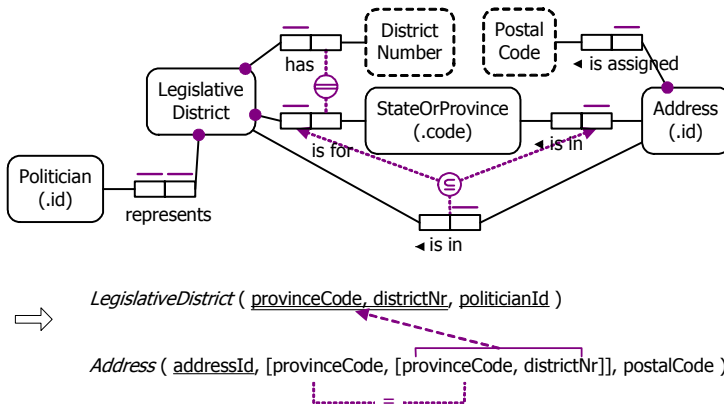


Fig. 1. A case of “redundant data” structure (from our previous ORM Workshop paper)

Note the “redundant” province-code column in the relational schema to which the ORM schema maps, using the standard Rmap algorithm (which applies no reduction transforms). The paper suggested the following schema as equivalent and preferable:

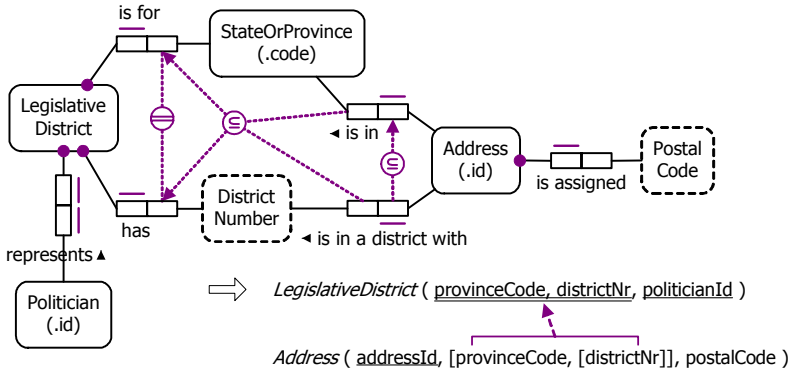


Fig. 2. An alternative ORM schema for the legislative-district example and its relational map

The object type to which is attributed what we might call the district-determining role, in the fact type Address(.id) is in [district-determiner], has changed: whereas it had been the LegislativeDistrict object type, the role is now played by DistrictNumber – which is a part of the reference scheme for the former, but only a part. This reduction transform, named “role redirection”, thus has the pattern shown in Fig. 3 [3, p. 706]:

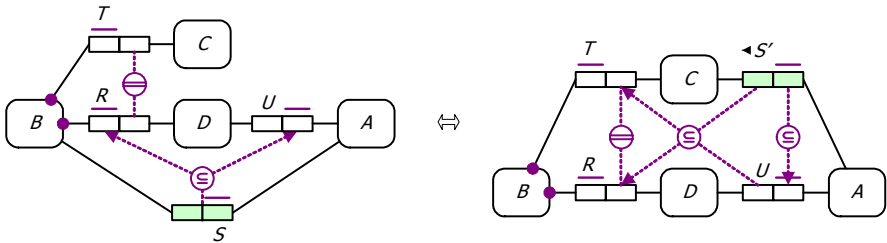


Fig. 3. A schema equivalence pattern underlying the general role redirection transform

The paper explored only “reduction transforms”, taking for granted such a conceptual schema; however, the fact that such redundancy-eliminating transforms are identifiable, raises other relevant questions such as, how does such redundancy arise? It arises of course from the conceptual schema; but how and why does it inhere in some conceptual schemata and not others? Is there an identifiable schema-characteristic which causes it to inhere? Is there an identifiable procedure by following which the problem may be avoided in the first instance? Or, on the contrary, is the occurrence or non-occurrence of such redundancy completely random, unpredictable, and (thus) unavoidable? Should its study thus be the job of statistics rather than of model-theory?

The latter may sound a bit far-fetched. We would suggest rather that there is indeed a procedure for determination of role-players in facts, by following which, all such anomalies of redundant data may be avoided in the first place, at the conceptual level.

The current paper suggests that the source of this phenomenon is lack of appreciation for, and of proper handling of, a particular aspect of the informal semantics being expressed by the information model, viz. whether a particular object reference is “information-bearing”; that is, poor decisions as to role-player depend upon such lack. The aim of the paper, which the later sections pursue, is to elaborate a solid theoretical basis and method for attribution of “roles” to objects where they are mentioned in particular facts. This elaboration draws on the sciences of lexical semantics, linguistic pragmatics, and communication theory (a.k.a. information theory): Section 2 analyzes the general meaning of “information-bearing” in this context – as compared and somewhat contrasted with its technical meaning in its provenance (viz. communication theory). Sections 3 and 4 explore the specific meanings of “information-bearing” for the cases of direct and indirect (anaphoric) references respectively, whether explicit or implied references. Section 5 outlines a procedure, based on these definitions, for determining properly whether mentioned objects are playing “roles” in the sample facts; “properly” here means, without loss of information yet also without introducing structures that, due to their informal semantics, bear non-informative data.

2 Defining Our General Notion of “Information-Bearing”

Conceptual modeling must, by definition, accurately reflect the informal semantics of the universe of discourse (UoD). This semantics includes the *purpose* of the statements, i.e., that which speech-act theory calls the *perlocutionary* meaning [1]. It follows, then, that conceptual modeling may, and should, take into consideration the “information-bearingness” of specific object-references in the sample facts grounding the conceptual schema, as this is one aspect of the statement’s purpose. For we are suggesting that payment of attention to this aspect is able, via a suitable modeling procedure, to prevent data structures with redundancies such as we have just seen. It is time now to define our general notion of “information-bearing” as used in this context.

The notion is derived more or less directly from a notion with the same name, found in information theory (a.k.a. communication theory). In this theory, the amount of “information”-content of a particular unit of expression is defined as a function of the unit’s *probability* of occurring. A special case of probability is *certainty*: a unit which cannot but occur in a given context, has there a probability of 1. The basic principle of information theory, from which we borrow, is this: the amount of information-content of an expression-unit is *inversely proportional* to its probability of occurring in its given context. In other words, if a particular expression-unit is certain to occur given the context, the expression-unit carries no information, and a loss (dropout) of just that expression-unit during the process of communication would not cause any loss of information, since the missing unit could be supplied based on its (successfully-transmitted) context [2, p. 169; 5, pp. 81-98].

That this notion may be applicable to the problem of “redundant data” seems readily apparent: in cases such as we have seen above, a whole column of data could be dropped from the database without any loss of information, and the reason for this

nonloss seems to be that, *in its given context* – which includes assumptions about the general scope and purpose of the database – *the column which we could thus drop could not have contained any different data-entries from those which it did*. We will apply this general idea, then, to choice of role-player objects: is the object's reference, at that place in the fact, "information-bearing" in this sense?

However, there are several clarification and qualifications which we must proffer, as to the application modelers should make of this principle. For example, we must clarify and show that this notion is properly informal and semantic, rather than formal and syntactic. Indeed, as we shall see, it applies just as much to implicit (unstated but intended) object references, as to explicit ones. Also, as we shall see, we must define the criterion of "information-bearing" somewhat differently for the indirect sort of references called *anaphoric*, than we must define it for normal, direct, object references. Finally, we must show that the crucial question, for determination of role-bearing, is more complex than just that of the reference's "information-bearing"-ness; we must determine its "*completely* information-bearing"-ness.

3 The Specific Definition of "Information-Bearing", as Applicable to Direct (Non-anaphoric) Object References

The simplest example of a non-information-bearing object reference is probably the mention, in a fact sample, of what we might call a scope-defining object. For purposes of an information system keeping records of Neumont University, for example, a sample fact's mention of the university itself would be a non-information-bearing reference: "The class HU110 at Neumont University is offered in quarters 1, 2, and 4." Given a context in which the scope of the universe of discourse is defined by the scope of that university, it is generally "redundant" to mention that university in a fact of interest (a fact within this scope). Attributing a role to the university in the above fact, would lead to the fact type `Class(code) at University(name) is offered in Quarter(nr)`, and the relational mapping would create a column that could never contain any data, on any row, but that references this very same university.

Note that this redundancy holds no matter how the university is referenced, and whether or not it is always referenced the same way: Although it is usual to assume there are no synonymous, different object references in the database, and absence of synonyms certainly makes the system easier and more efficient to query, conceptually the redundancy involved, in mention of the university in sample facts from the above domain, would still be involved in every such mention, even if the mentions used various synonyms: e.g. "Neumont", "N.U.", and "Neumont University." In other words, it is not the invariance of the "expression-unit" in this context, which makes it "non-information-bearing" for our purposes; rather, it is the invariance of the individual-concept intended by the expression(s), the invariance of object referenced. Thus, it is invariance of informal semantics, rather than of (formal) syntax.

The criterion of "information-bearing", for direct (non-anaphoric) references, thus is whether the reference, at that point in the fact-statement, could have been *to a different object*, without that change constituting the proposition expressed to be necessarily of a fact type outside the scope of the assumed universe of discourse – i.e., outside the

fact-domain of interest. Of course, we know not yet what the roles are, we are still in the process of determining that, and so have yet to determine the elementary “fact types” we want in our conceptual schema of the domain; but in the previous sentence, we used “fact type” in a looser sense than this, in which every mentioned object is considered a role-player at its point of mention, nor need the fact types be elementary. Thus, a non-anaphoric object reference is “information-bearing”, for our purposes, if and only if it could have been to a different object of the same type, without changing any of the *types* of objects mentioned in the fact statement, or the predicate text relating them, and the proposition if true would still be in scope, a fact of interest.

4 The Specific Definition of “Information-Bearing”, as Applicable to Indirect (Anaphoric) Object References

The sort of referential term which linguistic semanticists usually call an *anaphoric* term, is one whose definition involves (even if implicitly) “the linguistic, discourse-context” and the previous reference(s) of a particular sort found therein. Thus the term itself references objects indirectly, via a previous reference. The most common anaphoric terms, perhaps, are third-person pronouns (e.g. “she”, “he”, “it”, “her”, “his”, “its”; the latter two, of course, are possessive or otherwise genitive anaphoric terms, meaning “of him”, “of it”). Other common anaphoric references use a demonstrative pronoun: “*that* car” (i.e. the car previously mentioned; if the car is indicated by ostension instead, while saying “that car”, then the reference is *deictic*, not anaphoric; ostensive references, while interesting from a linguistic-pragmatical viewpoint, are irrelevant to information system design, being impossible in a database system).

Here is an example involving an anaphoric pronoun: “Fred Klett has spent 47 months outside the country of *his* birth.” Unpacking the meaning of this genitive term into an *of*-prepositional phrase, we might express the same fact this way: “Fred Klett has spent 47 months outside the country of birth of said person.” We have underlined here the intended *object terms* (i.e. references). As is standard for fully unpacked direct references to entities (rather than to values), the object reference to the country of birth contains one or more other object references within it; in this case, the contained reference (“said person”) is anaphoric. Given these object terms, the ORM schema for the domain of interest here would usually be the following:

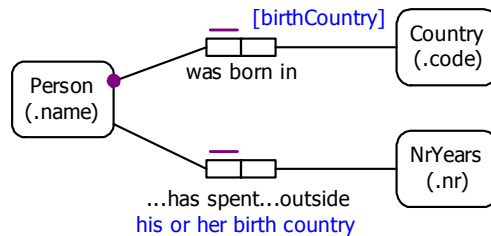


Fig. 4. An example ORM schema involving an anaphoric reference

Note that we have attributed no role, in the bottom fact type, either to the country reference or to the second, anaphoric, person reference. Clearly, attributing a role to either of these would create a structure for storing “redundant” data: the person in view is already playing a role in that fact, and it is precisely that previous mention via which the anaphoric reference refers; likewise the birth country is already recorded elsewhere for this person, in the top fact type. (In passing, it is interesting to note that it is apparently where there are anaphoric fact types of interest, that it sometimes seems more natural to objectify a fact type lacking a fully-spanning uniqueness constraint: for example, the fact type Moon orbits Planet seems to want objectifying, to the entity type “Orbit”, when there is also in this domain of interest the fact type Moon orbits *its planet* in NrDays; for one could then eliminate the relative awkwardness of an anaphoric reference (viz., its planet) by replacing the latter fact type with Orbit takes NrDays.)

So, what does being “information-bearing” entail for an anaphoric reference? It still entails that in that reference-occurrence’s linguistic context (including the domain’s scope-delimitation), it could have been referring to some other object than the one to which it does refer. But for anaphoric terms, this “context” includes also the “previous”, non-anaphoric reference *via which the anaphoric reference refers*. Therefore, since the anaphoric reference is by nature indirect, we must investigate whether, in a context *in which that previous reference is to the object x*, it is possible for the anaphoric reference to be changed, to refer (perhaps directly) *to some other object than x* (or rather: than whatever object that previous reference now references), without thereby excluding the fact from the domain-scope that contains all fact types of interest. If and only if that is possible, the anaphoric reference is information-bearing.

Let us make some clarifications at this point: Note that, as before, rather than on the (formal) *syntax* the question depends thus on the informal *semantics* – but, in this case, of both the current *and* the “previous” reference. Secondly, it is important to realize that the “previous reference” need not be, as here, to an object mentioned elsewhere in the database: as semanticists and linguistic pragmatists have noted, anaphoric references may be to something that is part of assumed, tacit knowledge common in the “domain of discourse” [5, p. 672; 3, p. 80n.]. Thus, a scope-defining object, such as “Neumont University” in our earlier example, could also be referenced anaphorically, as in “The class HU110 *in this university* (i.e. the university in view, the one delimiting this domain of discourse) is offered in quarters 1, 2, and 4.” The criterion for information-bearingness is the same for this as it is for any other anaphoric reference. Thirdly, note that the quality we have called “information-bearingness” is orthogonal to the quality we have called the “directness” of the reference: both anaphoric and direct object-references may be information-bearing or not.

However, there is another characteristic of object references, relevant to the determining of role-players, which is *not* orthogonal to the quality we call “information-bearing”; that is the quality we might call the explicitness (vs. implicitness, tacitness) of the reference. Only non-information-bearing references could be left tacit. And they often are. It is via the language science *linguistic pragmatics* that we learn why speakers and writers will often leave part of their intended sense unstated, and how that intended sense is recovered by the hearers or readers. As an illustration of one of the most important principles by which this takes place, let us return to the example cited above from last year’s ORM Workshop paper [3]. We repeat Fig. 1 (above) here for convenience:

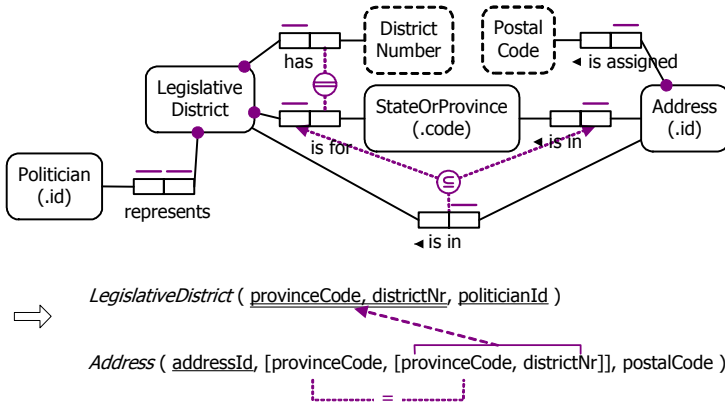


Fig. 5. Another case of “redundant data” structure (from our previous ORM Workshop paper)

Assuming this model was based on statements of sample facts, what might have been some typical fact statements for this domain? We would suggest that the following is a natural, likely statement of the sort(s) that might have been used:

“The address with the Id ‘1’ is in the state ‘AK’, in the legislative district with the number ‘2’.”

Since the scope of the UoD is so relevant, let us also stipulate that the scope here involves more than one state-or-province – which is why the schema indicates a legislative district must be identified compositely. Given that scope and that necessity for composite reference, however, there seems (on the surface anyway) to be an anomaly in the fact statement above: the phrase, “the legislative district with the number ‘2’”, is – at least, *as explicitly stated* – a *non-composite*, and thus *partial*, reference.

An anomaly like this, according to the “conversational *maxim of Relevance*”, would prompt the reader to look for some hidden, intended meaning that would make relevant what is explicitly stated. Among such conversational maxims, which, according to linguistic pragmatics, guide the formation and interpretation of discourse, is also another very important one [10, pp. 101-2]:

The maxim of Quantity

1. make your contribution as informative as is required for the current purposes of the exchange
2. do not make your contribution more informative than is required

The reader/hearer assumes that the conversational maxims are being honored by the speaker/writer. Therefore, when something about the communication *appears* to violate one of the maxims, the first reaction of the reader/hearer is to look for the manner in which the discourse’s form is actually in honor of one of the maxims.

Thus, in our example, it initially appears that the first part of the Quantity maxim is being violated, inasmuch as the definite article, *the*, in “the legislative district with the number ‘2’” signals an attempt to reference, to bring to the reader/hearer’s mind, one

particular legislative district, and yet not enough information is given, in that referential phrase, to identify a particular district within this domain's scope: we need also to know the state-or-province. The reader or hearer, therefore, looks for how the form of this contribution is actually necessitated by that maxim or by another. And the reader or hearer needn't look far, because the first part of the sentence has already implied the state-or-province of the legislative district: if an address is in a particular state, then obviously any legislative district it is in must be for that state – as is expressed by the join-subset constraint in the ORM schema. Thus, the second part of the Quantity maxim is actually being honored, by the above seeming anomaly in this discourse-contribution's form. The intended reference is in fact composite, though partly implicit; and could be made explicit fairly naturally (if rather redundantly) by explication of the (anaphoric) reference to the state-or-province: “The address with the Id ‘1’ is in the state ‘AK’, in the legislative district (which is in *that address's state* and) which has the number ‘2’.” Again, the fact that this reference was left unstated (though intended) shows clearly that it is a non-information-bearing reference: it is not possible for a tacit reference to “bear information” in our sense of the term, else its information could not be inferred, and thus recovered, from the context, as is doable in this example.

5 How to Determine Whether Mentioned Objects Are “Role-Players” at the Point of Their Mention in the Sample Fact

Now, unlike this last example, we have seen a couple of previous examples in which the non-information-bearing object term is one that is *not* an element within another one; thus we might call it a *maximal* (i.e., non-included) object term. As we might glean from those previous examples, an object referenced by a *non*-information-bearing *maximal* term, should not be considered a role-player at that point in that fact, lest redundant data structures be introduced into the schema.

But what about our last example, where a non-information-bearing (indeed, in this case, tacit) object term *is* an element within another object term? Should we, in such a situation, ever consider the object that is referenced by the *containing* object term a role-player in the fact? Let us look at this example again, with each of its maximal object terms underlined, and summarize what we have learned about it:

“The address with the Id ‘1’ is in the state ‘AK’, in the legislative district (which is in *that address's state* and) which has the number ‘2’.”

Here is what we have elucidated, so far, about the maximal object references in this statement: The first two are simple (non-composite), direct (non-anaphoric), and information-bearing (according to our above definition). The third maximal reference, to the legislative district, is direct, information-bearing, and composite – containing within itself two references (to *other* objects), the first of which (“that address's state”, i.e. “the state of that address”) is anaphoric and non-information-bearing, and the second of which (“the number ‘2’”) is direct and information-bearing.

Now, even though the reference to the legislative district is information-bearing (it could have been to a different district), brief reflection will indicate that attribution of

a role to this legislative district, at this point in the fact, would create a data structure that is at least partially redundant. (It is precisely this that led to the redundant column in the relational mapping in Fig. 5.) Therefore, it is not enough to say that in order to reference a role-player, an object term must be information-bearing; it must be something further, which we might call “*completely* information-bearing”. We could define the latter characteristic as follows (when we say “term” we mean its occurrence):

completely information-bearing object term:

an object term which is information-bearing and which, in the object-term containment-hierarchy implicit in its fact-statement, is either a leaf node, or else contains only nodes each of which is a *completely information-bearing object term*

“Leaf node” means a non-containing object term. Note that the definition is recursive: we must know whether a contained term is “completely information-bearing”, before we can know whether the term that contains it is. Also note that, as one might assume, an object term cannot be *completely* information-bearing if it is not information-bearing. (This qualification becomes crucial if any object-synonyms are allowed.)

This definition, and its recursive nature, give rise to the following necessary procedure for determining which object-references should be considered as being of role-playing objects (if we wish to avoid redundant object-references in our database):

1. Expose the referential structure, the containment-hierarchy tree, within every object term in the sample fact, recursively identifying all contained terms;
2. within each such hierarchy identify each *completely information-bearing* object term (which must be done starting with the leaf-nodes in the hierarchy);
3. each *maximal* such object-term (i.e., each one not contained within another such term), should be considered a reference to a *role-playing* object.

Step 1’s exposure of the structure of the referential meaning, hierarchically, is best done outside the sentence; while listing references, unpack direct references into “definite descriptions”, and anaphoric references into phrases of the form “the <object type name> already/elsewhere mentioned”. Then, looking within these unpacked references, again apply the procedure, recursively. To illustrate with our sentence:

“The address with the Id ‘1’ is in the state ‘AK’, in the legislative district (which is in that address’s state and) which has the number ‘2’.”

- *the address with the Id ‘1’
 - *the address-Id ‘1’
- *the state with the code ‘AK’
 - *the state-code ‘AK’
- the legislative district that is in that address’s state and has the number ‘2’
 - the state of that address
 - the address already/elsewhere mentioned
 - *the district-number ‘2’

After listing the referential structure in this way, one does step 2, identifying all completely information-bearing object terms, starting with the leaf nodes. Such terms should be marked, as we did with asterisks (*) in the above tree. The role-playing objects will be those with *maximal completely information-bearing* references (i.e., such references that are not contained in any other such references). Each role-player's reference should then be marked as such, after being unpacked now in the original sentence *as a definite description* (even if that reference was anaphoric). The chosen manner of marking the role-playing objects' references might depend on the language in which one is working: in English, one could simply capitalize the name of the role-playing object's type; but in German, where all nouns are capitalized already, another manner of marking the reference must be used (e.g. capitalizing all letters of the object type's name). Thus the result of this procedure, for the current example, yields:

“The Address with the Id ‘1’ is in the State ‘AK’, in the legislative district with the DistrictNumber ‘2’.”

Here our marking-via-capitalization makes clear that the objects that are playing roles in this fact are the address, the state, and the district-number—but *not* the district itself. In moving to fact *types*, references to non-role-players should be left as predicate text.

Questions remain as to where, in the conceptual schema design procedure, this role-player-determining process should occur, and how well it might be adapted to a schema design procedure that focuses first on identifying fact types rather than fact instances. On the latter question it must be emphasized that the two criteria of “information-bearing” we have identified both involve the variability of *object referenced*. It therefore seems impractical to attempt this role-player-determining procedure on fact types rather than on fact samples. Ultimately only fact samples justify fact types.

On the other question, as to where in the schema design procedure one should determine role-players, it seems to us that this must be done prior to elementarizing the fact-samples. The reason is that “elementary facts”, as we usually understand this term, must (in addition to being non-conjunctive) always have at least one role-playing object. If the above process shows, therefore, that no objects are playing roles in the fact, we must disqualify the latter, subsequently, from the status of elementary fact—and not create any fact type, based upon it, in our schema design.

References

1. Bach, K.: Speech Acts and Pragmatics. In: Devitt, M., Hanley, R. (eds.) Blackwell Guide to the Philosophy of Language (2003), <http://userwww.sfsu.edu/~kbach/Spch.Prag.htm>
2. Eco, U.: Semiotics and the Philosophy of Language. Indiana University Press, Bloomington (1984)
3. Halpin, T., Carver, A., Owen, K.: Reduction Transforms in ORM. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 699–708. Springer, Heidelberg (2007)
4. Levinson, S.C.: Pragmatics. Cambridge University Press, Cambridge (1983)
5. Lyons, J.: Introduction to Theoretical Linguistics. Cambridge University Press, Cambridge (1968)
6. Lyons, J.: Semantics, vol. 2. Cambridge University Press, Cambridge (1977)

A Closer Look at the Join-Equality Constraint

Gerhard Skagestein and Ragnar Normann

University of Oslo, Department of Informatics,
PO Box 1080 Blindern, NO-0316 Oslo, Norway

Abstract. Assume an ORM-diagram where there exist two (or more) paths from one entity type to another through a sequence of many-to-one binary facts. Here we may have a join-equality constraint saying that if we follow the different paths from an instance of the first entity type, we should find the same instance of the second entity type. This constraint is inherent in most ticketing/reservation systems, where it may give rise to overlapping foreign keys in the relational database. Another interesting observation is that if a relation is in 3NF, but not in BCNF, there must be a join-equality constraint in the underlying model.

Keywords: Join-equality constraint, Overlapping foreign keys, Boyce-Codd Normal Form.

1 Introduction

Assume that we have a fact-oriented model like the one in Fig. 1. In a Location (theatre, concert hall, sport-stadium, etc.) there are several Seats. In such a Location, several Events may take place – probably not overlapping in time. For each Event, Tickets (or reservations) will be issued; each Ticket will give the right to use a certain Seat on a certain Event. In all of these kinds of models, we have a join-equality constraint capturing the rule that a Ticket must be to a Seat that is situated in the same Location as where the Event specified on the Ticket is taking place. (You could argue from the model that you can do without the fact type between Event and Location because you can find a Location for an Event through one of the Tickets and Seats for that Event, but you probably will need to populate the fact type before any Ticket is issued at all. A similar reasoning applies for the fact type between Seat and Location.) If the Location is a singleton, or left out of the model, the join-equality constraint is of course always satisfied trivially. Generally, we very often see the join-equality constraint in fact-oriented models encompassing sets of reusable resources subject to reservations, logging or ticketing.

The need for the join-equality constraint in such models was discovered by data modellers in Norway and the Netherlands almost simultaneously around 1985, and discussed informally in meetings organized by the consulting branch of the Control Data Corporation. The constraint was at that time given the name “the equivalence-of-path constraint”, but, to our knowledge, nothing was then published about it internationally. The constraint is, however, mentioned in a

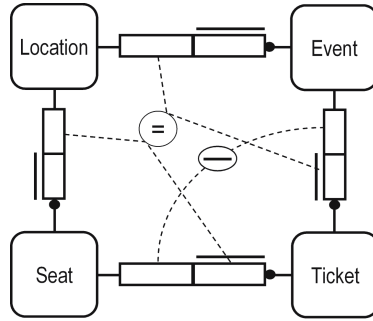


Fig. 1. The Ticket-model and the join-equality constraint

Norwegian data modeling exercise book from 1987 [8] and devoted a subsection in a Norwegian textbook on data modelling from 1991 [7].

Since then, the general theory on join constraints has been developed [5,6]. Most examples, however, concern the external uniqueness constraint and the join-subset constraint. The join-equality constraint is briefly mentioned on page 405 in Halpin and Morgans newest book on Information Modeling and Relational Databases [6], but is there claimed to be “less common”. We think, however, that the join-equality constraint deserves a little more attention. The constraint is essential in certain kind of models, and it can give us a clearer view on some overlapping foreign keys. We also show that if a relation is in 3NF, but not in BCNF (about normalization, see for instance [4, Chapter 10]), there must be a join-equality constraint in the underlying model.

2 Enforcing the Join-Equality Constraint in a Relational Database

When we map the model in Fig. 1 into a normalized relational database schema by means of the Rmap procedure (see [6]), we get the structure shown in fig. 2. Enforcing the join-equality constraint will involve two join-operations or two look-up-operations to be able to check that the two Location-attributes really have the same value in all instances. In order to avoid these joins or look-ups every time a Ticket is inserted or changed, most database designers would probably go for the 1NF structure shown in Fig. 3. Since the join-equality constraint dictates the values of location1 and location2 to be the same for all instances, the most obvious implementation is to replace the two columns by one single, common column, as shown in Fig. 4. This is the design that most ER-modellers probably would come up with offhand, without even knowing of anything like the join-equality constraint.

A common variant of such models is to include the Location in the reference to Seat so that we get a partly information bearing Seat identifier. Then, one of the Location-attributes will be mapped into the Ticket-relation even in 3NF,

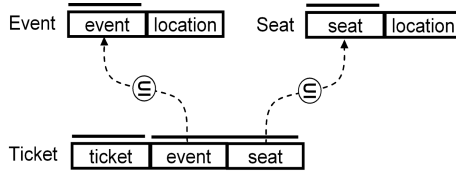


Fig. 2. The Ticket-model grouped to a normalised relational database schema

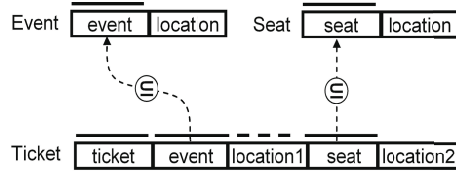


Fig. 3. The Ticket-model grouped to a 1NF relational database schema

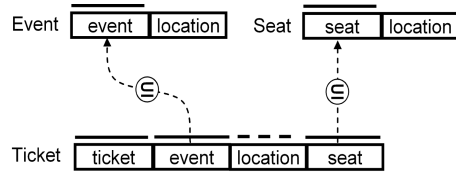


Fig. 4. Implementation of the join-equality constraint

so that only one join-operation or look-up-operation is required to check the join-equality constraint. But again, in order to avoid the join during runtime, most database designers would probably go for a structure similar to the one in Fig. 4.

3 Overlapping Foreign Keys

Let us now turn our attention to a variant of the example model in which the Location is part of the reference both to Event and Seat. These kinds of models are very often found in reservation/ticketing systems for transportation companies. To make our example more concrete, we let the Location be a passenger ferry – in the model called Ship –, the Event will be the Departure of the Ship from the harbour, and the Seat will be a Berth in a cabin on the Ship. The model is shown in Fig. 5 (We assume here that a Ship will have no more than one departure per day – if that is not the case, we have to employ a time-axis with a finer granularity.) The join-equality constraint now states that a Ticket to a Berth on a Ship must be to a Departure that is serviced by the very same Ship.

The Rmap-procedure applied to this model will give the relational database schema depicted in Fig. 6. Because of the compound reference schemes for both

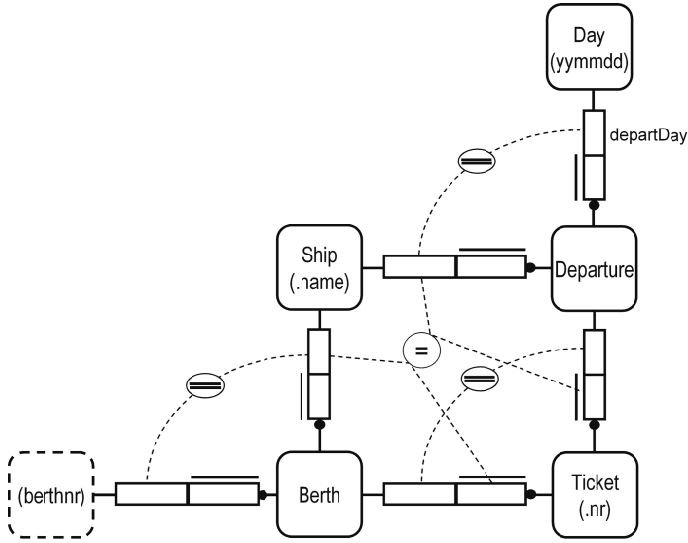


Fig. 5. The join-equality constraint in the transport company model

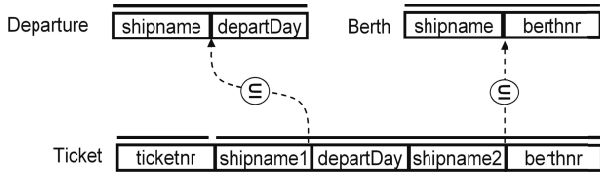


Fig. 6. The relational database schema for the transport company

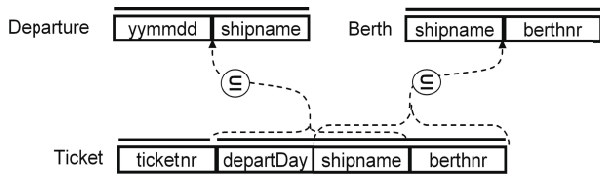


Fig. 7. Implementation of the join-equality constraint

Departure and Berth, the two shipname-attributes will appear in the Ticket-relation even in 3NF. Again, we can implement the join-equality constraint just by replacing the two shipname-attributes by one common attribute as shown in Fig. 7. The interesting side-effect of this little trick is that the two foreign keys now will overlap, because they will have the attribute shipname in common.

In his book *Relation Database, Writings 1985-1989* [3], Part I, chapter 18 “Why overlapping keys should be treated with caution”, Chris Date warns about

partly overlapping foreign keys, claiming that they will introduce dependencies in the structure and cause side-effects when running the database. His example has the same structure as ours; however, Location, Departure, Berth and Ticket are replaced by Location, Department, Project and Employee, and the business rule giving rise to the join-equality constraint is “If employee *e* works for department *d* at location *l1* and also works on project *j* at location *l2*, then *l1* and *l2* must be the same location”. If we choose to propagate to the Employee relation a change or delete in the Department relation in order to keep the referential integrity constraint satisfied, we also have to think about how to keep the referential integrity constraint between Project and Employee satisfied, and we probably end up with a side-effect. Furthermore, if some day the join-equality constraint no longer holds because of changes in the business rules, the database has to be redesigned by again splitting the location attribute in the Employee relation.

However, Dates concerns are fuelled by his (deliberately?) chosen Universe of Discourse (UoD), in which the stability of the join-equality constraint is questionable. In the transport-business UoD, the join-equality constraint is an inherent truth, and no database programmer would even think of propagating a delete of a departure to all the tickets automatically – there are some passengers around that should be notified and have their tickets changed.

4 The Join-Equality Constraint and the 3NF/BCNF-Distinction

Let us look at the example in Fig. 8. Customers are related to Departments (of a company), and for each such CustomerRelation, there is a Employee responsible. An Employee is working for a Department. The join-equality constraint is saying that for a CustomerRelation, the Employee responsible must work for the Department of that CustomerRelation, and, at the same time, an Employee must be responsible for a CustomerRelation to be allowed to work in the Department in question.

If we now group this model to a 2NF structure by joining in the employee-department fact in order to avoid joins during runtime, we get the relation shown in Fig. 9. Functional dependencies are shown by arrows. (Replacing the join-equality constraint with a join-subset constraint may give a more sensible UoD, but if we then group the model the same way, the resulting relational

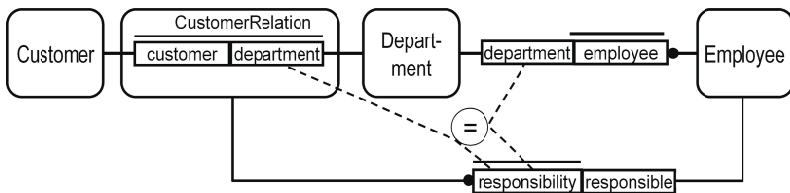


Fig. 8. A model that may give a relation in 3NF which is not in BCNF

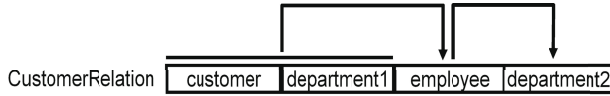


Fig. 9. The model grouped to a 2NF relational database schema

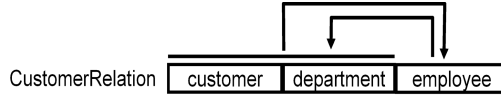


Fig. 10. Implementation of the join-equality constraint violates BCNF

database schema will violate the Entity integrity rule, which says that no part of a primary key can be null [2].)

Observing that the join-equality constraint will force the values of department1 and department2 to be the same for all instances of CustomerRelation, we may as before implement the constraint just by replacing the two department attributes by one common attribute. We then get the relation schema shown in Fig. 10, which is in 3NF, but not in BCNF [4].

This is an example of the following observation: Whenever we have a relation that is in 3NF, but not in BCNF, there must be a join-equality constraint in the underlying model.

Proof. In a relation satisfying BCNF, all non-trivial functional dependencies (FDs) $X \rightarrow A$ must have a superkey as its left hand side X . In 3NF we also allow all FDs $X \rightarrow A$ where A is a key attribute, i.e., $A \in K$ where K is a candidate key.

Thus, in any 3NF-relation R that is not in BCNF, there must be a nontrivial FD $X \rightarrow A$ (i.e., $A \notin X$) where X is not a superkey, and $A \in K$, a candidate key. Furthermore, the FD $K \rightarrow X$ cannot be trivial (as this would make $K \setminus A$ a key, violating the minimality of the candidate key K).

We then have the trivial FD $K \rightarrow A$ and the two non-trivial FDs $K \rightarrow X \rightarrow A$. In any ORM-diagram having R as (part of) its mapped result, these FDs will show up as a join-equality constraint from K to A . \square

5 Summary

The join-equality constraint very often shows up in fact-oriented models encompassing sets of reusable resources subject to reservations, logging or ticketing. The join-equality constraint may be implemented by replacing two attributes with one common attribute, if the model is mapped in such a way that the two attributes are in the same relation. If partly information-bearing identifiers are used in the model, this may give rise to overlapping foreign keys – an overlapping that can be considered safe and sound.

Furthermore, we have shown that whenever we have a relation that is in 3NF, but not in BCNF, there must be a join-equality constraint in the underlying model.

References

1. Codd, E.F.: Recent investigations in Relational Database Systems. In: Proceedings of the IFIP Congress (1974)
2. Codd, E.F.: Extending the Database Relational Model to Capture More Meaning. *ACM Transactions on Database Systems* 4(4) (1979)
3. Date, C.: *Relational Database, Writings*. Addison Wesley, Reading (1990)
4. Elmasri, R., Navathe, S.B.: *Fundamentals of Database Systems*. Pearson/Addison Wesley, Boston (2007)
5. Halpin, T.: Join Constraints, <http://www.orm.net/pdf/JoinConstraints.pdf>
6. Halpin, T., Morgan, T.: *Information Modeling and Relational Databases*, 2nd edn. Morgan Kaufmann Publishers, San Francisco (2008)
7. Skagestein, G.: *Data i fokus*, Universitetsforlaget, Oslo (1991)
8. Skagestein, G., Thorvaldsen, A.: *Arbeidshefte til Fra virkelighet til datamodell*, Universitetsforlaget, Oslo (1987)

Model Ontological Commitments Using ORM⁺ in T-Lex

Yan Tang and Damien Trog

Semantic Technology and Application Research Laboratory (STARLab),
Department of Computer Science, Vrije Universiteit Brussel,
Pleinlaan 2, 1050 Brussel, Belgium
{yan.tang, Damien.trog}@vub.ac.be

Abstract. When designing and developing ontology based applications, we semantically ground them by ontologically committing the application rules to their respective domain. These rules can be, for instance decision rules for a decision support system. For the DOGMA framework we have introduced ORM⁺, a novel extension of ORM for modeling, visualizing and interchanging ontological commitments. In this paper, we illustrate our ongoing research on ORM⁺ and T-Lex as its supporting tool. We demonstrate in the field of on-line customer management.

1 Introduction

An ontology is a semiotic representation of agreed conceptualization in a subject domain [1, 8]. As an ontology modeling framework and ontology engineering methodology, DOGMA (Developing Ontology-Grounded Methods and Applications, [14, 18]) was designed and inspired by the tried-and-tested principles from conceptual database modeling. In DOGMA, an ontology is represented in two layers using the *double-articulation* principle: the *lexon* layer and the *commitment* layer.

A *lexon* is modeled as a quintuple $\langle \mathcal{Y}, t_1, r_1, r_2, t_2 \rangle$, where t_1 and t_2 are terms that represent two concepts in some language. r_1, r_2 are roles (r_1 corresponds to “role” and r_2 - “co-role”) referring to the relationships that the concepts share with respect to one another. \mathcal{Y} is a context identifier. It is assumed to point to a resource, usually a document in the general sense, where the terms t_1, t_2 are originally defined and disambiguated, and in which the roles r_1, r_2 become “meaningful”. For example, a *lexon* $\langle \mathcal{Y}, \text{order manager}, \text{accept}, \text{is accepted by}, \text{customer request} \rangle$ explains a fact that “order manager accepts customer request”.

An ontology is often designed for several tasks or applications. As Guarino mentioned, the value of an ontology is the reusability in the context of knowledge management [8]. In order to ensure the reusability of an ontology, the *commitment* layer in DOGMA is designed to separate the application layer and the *lexon* layer. It contains a set of ontological commitments, with which an application commits its local vocabulary and application semiotics to the meaning of the ontology vocabulary.

The commitments need to be expressed in a *commitment language* that can be easily interpreted by the domain experts and a machine. In [1, 18], the authors studied many advantages of using ORM (Object Role Modeling, [9]) as a commitment modeling manner. Furthermore, they argue that it is rather feasible to model and visualize

the commitments using ORM, for ORM has an expressive capability in its graphical notations and verbalization possibilities. The ORM commitment models can be stored in XML files (ORM Markup Language, ORM-ML for short, [1]), with which agents can share the ontologies.

Suppose that a domain expert wants to constrain the lexon *<Customer Request, is Accepted By, accept, Order Manager>* with a uniqueness constraint like “one customer request is accepted by *at most one* order manager”. Its ORM diagram is shown in Fig. 1.

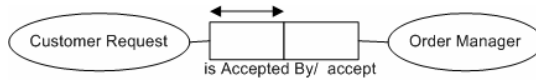


Fig. 1. Model ORM uniqueness constraint

The corresponding XML file is shown as below:

```

...
<Predicate id= "lexon-1">
  <Object_Role ID= "lexon1_forward"
  Object="CustomerRequest" Role="isAcceptedBy" />
  <Object_Role ID= "lexon1_backward"
  Object="OrderManager" Role= "accept" />
...
<Constraint xsi:type= "Uniqueness">
  <Object_Role>lexon1_forward</Object_Role>
</Constraint>
...

```

ORM, ORM 2¹ and ORM-ML are the technology that is mainly used in DOGMA. However, it still lacks several operators and connectors while grounding the semantics for the application rules, e.g. the *sequences* and *dependences*. Moreover, ORM has certain limitations on using some specific operators, such as the *implication* operator. As the response, we have proposed ORM⁺ (an extension to ORM) and ORM⁺ ML (a hybrid language of ORM-ML and FOL Rule ML) in our early paper [21]. Seven new graphic notations, including *negation*, *conjunction* and *sequence* operators, were introduced. These notations are mainly used for modeling commitments used by ontology based decision support systems, such as Semantic Decision Tables [22].

Recently, we have been working on a modeling tool called T-Lex [23] for modeling ontological commitments using ORM⁺. T-Lex is a tool for graphical ontology engineering that has a notation based on ORM.

By using T-Lex, the most significant constraints for ontologies (mainly based on ORM) are supported [24], which include most database consistency constraints. Because of the tree representation, constraints are indexed instead of connected with a line to avoid cluttering.

In this paper, we focus on the ongoing researches on ORM⁺ and discuss how to model, visualize and verbalize the ORM⁺ models in T-Lex. The rest of the paper is

¹ ORM 2 [11] is the second generation of ORM. Its graphical notations are improved based on industrial experiences.

organized as follows: section 2 is the related work. We compare our work to other ontology modeling tools, such as Protégé. Seven graphical notations are demonstrated in section 3. We conclude, open a discussion and discuss the future work in section 4.

2 Related Work

The current state of the art on ORM modeling is the NORMA tool from Neumont University [1]. It supports the new ORM2 notation and is able to map the schemas to the popular RDBMs.

Protégé [16] is an ontology development tool developed at Stanford University. For Protégé different visualization modules have been developed, with Jambalaya [19] as the most popular one. It uses a nested graph-based representation of hierarchical structures, together with nested interchangeable views.

Variations of the spring embedded algorithm [3] are also widely used in ontology engineering tools. Examples are the work of Mutton and Golbeck [15], the OIModeler plug-in for the KAON server [5], and the visualization user interface in OntoEdit [20]. The ontology is considered as a graph whose vertices represent concepts and whose edges represent relationships. Vertices are positioned randomly as their initial position. Each vertex is considered to cause a repulsive force on the others, while edges represent an attracting force. When minimum energy is reached, the visualization is complete. The advantage is that high-level structure can be detected. The disadvantage is that the number of iterations and the initial positions can create a new representation on each visualization.

Another approach is using cluster maps for visualizing populated, light-weight ontologies [4]. This visualizes the instances of a number of selected classes from a hierarchy, organized by the taxonomy.

Pretorius also visualized the lexon base by employing a fish eye view on an ordered visual representation [17]. The technique is well suited for an overview of very large lexon bases, but not for searching for particular terms.

An overview of other ontology tools can be found in several surveys [13, 6].

Unlike all the tools illustrated above, T-Lex [23] uses *NORM* tree to tackle the lexon base visualization problems in a *scalable* and *structured* manner. *NORM* is a recursive acronym of “*NORM* Ontology Representation Method”. It is a method to represent domain ontologies in an undirected rooted tree format. T-Lex is particularly suited for small to medium sized ontologies. This is partly supported by grouping lexons by context and by the assumption that the user already knows the starting point for searching. Just like the term *NORM* is a recursive definition, the *NORM* tree in general is recursive. It can be spanned infinitely. By using T-Lex, the users can have *more flexible* ontology views than other ontology modeling tools.

3 ORM⁺ in T-Lex

In [21], we introduced the ORM⁺ graphical notations of *negation*, *conjunction*, *implication*, *sequence*, *necessity* and *possibility*. Negation, conjunction and implication are

borrowed from the basic operators² of propositional logics. Sequence is used in the commitments related to the time issue. The necessity and possibility operators are two basic ones in Modal logic. We illustrate their graphical notations in the following subsections.

3.1 Negation

It is possible to model a negation constraint using ORM. In ORM, one uses “closed-world” and specific “open-world” assumptions [9, pp. 61]. The “closed-world” assumption uses the *absence* of positive information (e.g. Customer request is accepted by Order manager) to imply the negative (e.g. Customer request is not accepted by Order manager). With an “open-world” approach, negative information is explicitly stored using negative predicates or status object types. I.e. “Customer request” has Acceptance status {‘Accepted’, ‘Not Accepted’} and *each* “Customer request” has *at most one* Acceptance status (Fig. 2). Or, “Customer request” has two subtypes “Accepted customer request” and “Unaccepted customer request”, which are mutually exclusive (Fig. 3).

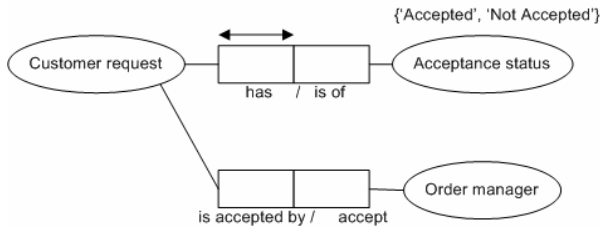


Fig. 2. Model negation in ORM – method 1

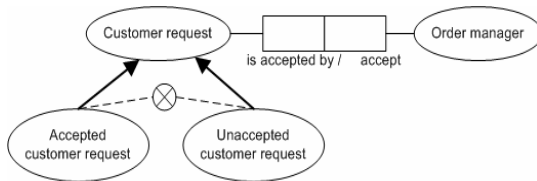


Fig. 3. Model negation in ORM – method 2

Although transferring the negation connective to the value constraint or the exclusive constraint doesn’t lose any information of a negative proposition, it is still not easy for a domain expert to know when the negative status is taken. For both positive and negative statuses of a type are modeled in the same schema. The domain expert

² In propositional logic, there are five basic operators and connectors: *negation*, *conjunction*, *disjunction*, *implication* and *equivalence*. The disjunction can be modeled using the exclusive-or constraint in ORM. The *equivalence* can be modeled using the *equality* constraint in ORM. Therefore, the equivalence and disjunction are excluded in the paper.

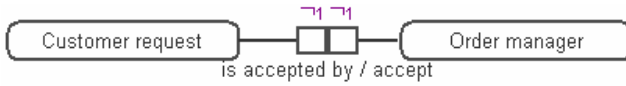


Fig. 4. Model ORM⁺ negation in T-Lex

has to make the analysis or reckon on extra information in order to know whether he uses the negative status or the positive one. To simplify the situation, we introduce the constraint of negation in ORM⁺.

Fig. 4 shows an ORM⁺ diagram containing a notion of negation. A “¬” is marked above each applied role. The number (e.g. ‘1’ in Fig. 4) after “¬” indicates the constraint group. Note that a negation constraint needs to be applied to both role and co-role. Because the lexon role and co-role often has the same meaning, e.g. ‘is accepted by’ and ‘accept’ in Fig. 4. It is strange for a role to have a negative situation while the co-role stays positive, and vice versa.

Fig. 4 is verbalized as “a customer request is not accepted by an order manager; an order manager doesn’t accept a customer request”. When an application receives this commitment, it will give two resulting sets: one contains the customer requests that are not accepted by any order managers; the other contains the order managers that do not accept any customer requests.

The negation is often used with other connectors, such as the implication, which will be discussed later.

3.2 Conjunction

The *conjunction* binary operator is to construct a logical AND with the conjunction operator \wedge . The conjunction operator is equivalently used as *set intersection* in the set theory. It is very useful to restrict the population of an object that plays a specific role. One writes such ontological commitments at the query level, e.g. list all the customers who are listed in a customer catalog AND whose state is normal”.

ORM doesn’t provide modeling techniques for the conjunction operator. In ORM⁺, a “ \wedge ” denotes the conjunction operator.

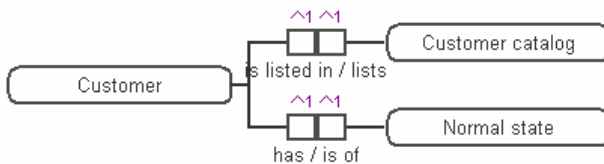


Fig. 5. Model ORM⁺ conjunction in T-Lex

Fig. 5 is an example containing a conjunction connector. We verbalize it as “a customer is listed in a customer catalog AND he has a normal state”. Note that the conjunction connective is often applied to more than two lexons. It is not possible to be applied on only one lexon.

When a decision supporting system receives an ontological commitment shown in Fig. 5, it will give a record of ‘customer’ that is listed in a ‘customer catalog’ and he has a ‘normal state’. As the negation operator, the conjunction operator is also often used with the implication connective.

3.3 Implication

In ORM, the implication operator is modeled using a *subset* operator. Fig. 6 shows an ontological commitment that the set of the members of ‘Driver’ who has ‘Driver’s license’ is the subset of the members of ‘Driver’ who has ‘License’. ORM uses a dotted arrow-tipped bar running from the subset role to the superset role for the subset constraint. It is comparable to the logical connective \rightarrow , which can be verbalized as “IF..., THEN” alike sentences. For example, Fig. 6 is verbalized as “IF a driver has driver’s license, THEN he has license”.

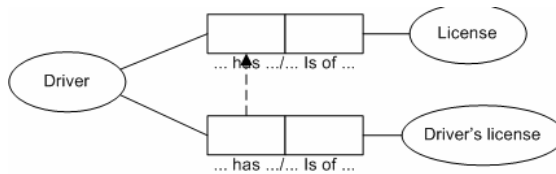


Fig. 6. Model implication using ORM subset constraint

However, there are two limitations while modeling the implications using ORM. One is that users can only model *monotonic* rules in ORM. In practice, we often encounter non-monotonic rules. E.g., the ontological commitments for the ontology based decision support systems often include *non-monotonic* rules, which are verbalized as “IF...THEN...ELSE” alike sentences and cannot be modeled using ORM.

The other limitation is, ORM implication constraint can only be applied to one object type (lexon term), such as “Driver” in Fig. 6. It is normal to model ORM implication in this way, because the ORM subset (implication) constraint is initially used for *set comparison* instead of *logical reasoning*. We need a more powerful notation to model event-driven decision commitments.

We use the symbol \Rightarrow to designate the *implication* logical operator in the ORM⁺ diagram (Fig. 7). The condition is indicated with \Rightarrow followed by a group number and the action is illustrated by \Rightarrow started with a group number. We verbalize Fig. 7 as: *IF* a customer is *NOT* listed in a customer catalog (a customer catalog does *NOT* list a customer), *THEN* an order manager creates a new customer.

Fig. 7 is a simple example that consists of a monotonic decision rule. We are able to model as well non-monotonic decision rules using ORM⁺. Fig. 8, for example, is verbalized as “*IF* a customer is *NOT* listed in a customer catalog, *THEN* an order manager creates a new customer, *ELSE* the order manager approves the customer request.” Note that the group number is very important in T-Lex. It categorizes the modeling information. For example in Fig. 7 and Fig. 8, the negation operator is grouped together with the action “an order manager creates a new customer”.



Fig. 7. Model ORM⁺ implication in T-Lex

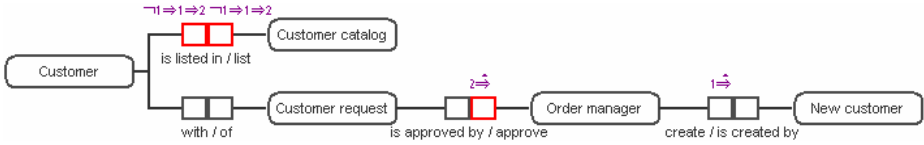


Fig. 8. Model non-monotonic decision rules in T-Lex

3.4 Sequence

The *sequence* operator in ORM⁺ is application oriented. ORM doesn't provide modeling methods for this kind of operators.

The definitions of the sequence constraint may differ between domains [21]. However, the core message is the same. That is, the issue of *order*, regardless of in the measure of time or space. We intend to use the sequence operator to reason on *orders*, e.g. the execution order of processes. Suppose that we have a rule “an order manager verifies a customer request *AFTER* the order manager receives the customer request”, which constrains the execution order of two processes.

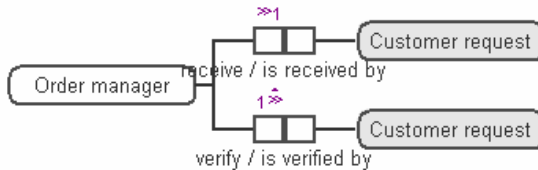


Fig. 9. Model ORM⁺ sequence in T-Lex

The event that happens earlier is indicated with a \gg followed by a group number. The one that happens later is marked with a \gg after the group number.

3.5 Other ORM⁺ Graphic Notations in T-Lex

With regard to other ORM⁺ graphic notations, there are two important operators in the Modal Logic – the Necessity and Possibility. In the research field of object-role molding, Halpin categorizes rule modalities into *alethic* and *deontic* [10]. Alethic rules “impose necessities, which cannot be violated by any applications because of physical or logical law...”, e.g. there is only one sun in the sky. A deontic rule

“imposes obligations, which may be violated, even though they ought not ...”, e.g. no one is allowed to kill another person.

In ORM 2 [11], an alethic modality of *necessity* \square is used for positive verbalizations by default. For example, the fact ‘a customer is listed in a customer catalog’ may be explicitly verbalized as ‘a customer is NECESSARILY listed in a customer catalog’ by default. Halpin interprets it in terms of *possible world semantics*, which are introduced by Saul Kripke et al. in the 50’s [12]. A proposition is “*necessarily true if and only if it is true in all possible worlds*”. The *facts* and *static constraints* belong to a possible world, in which they *must* exist at some point in time. Therefore, the necessity operator may explicitly append on the fact ‘a customer is listed in a customer catalog’ by default.

The necessity and possibility operators are important in the decision support world, e.g. in e-court. Therefore, their graphic notations are explicitly introduced. The necessity constraint is indicated with \square above the applied roles (Fig. 10).



Fig. 10. Model ORM⁺ necessity in T-Lex

4 Conclusion, Discussion and Future Work

In this paper, we have discussed our ongoing work on ORM⁺, which is an extension to ORM, and T-Lex as its supporting tool. The work is based on our experience on modeling ontological commitments for decision support.

As Halpin discussed, there are, in principle, *infinitely* many kinds of constraints [9, pp. 16]. This principle is general. It is not only for ORM, but also for many other modeling languages. As new problems bring forward new needs, one can always extend a modeling tool.

However, the modeling means will be more and more complicated until we cannot handle it. Therefore, we need to be very careful when we introduce new notations. Before extending an existing modeling tool, the following questions need to be answered: 1) Can the new constraint be modeled using a combination of existing constraints? 2) Is this new constraint really useful? How many applications will use it? In our problem settings, the graphical notations introduced in this paper are mainly used in the context of ontology-based decision support systems. If our applications are relevant to decision support, the above two questions can be answered.

Currently, we get more and more requirements on reasoning on these notations. In the future, we’ll focus on the reasoning issue of ORM⁺.

Acknowledgement. We’re pleased to thank Gu Yan for programming. The research is partly supported by EC Prolix project.

References

1. Curland, M., Halpin, T.: Model Driven Development with NORMA. In: Proc. 40th Int. Conf. on System Sciences (HICSS-40). CD-ROM, p. 10. IEEE Computer Society, Los Alamitos (2007)
2. Demey, J., Jarrar, M., Meersman, R.: Markup Language for ORM Business Rules. In: Proc. Of International Workshop on Rule Markup Languages for Business Rules on the Semantic Web (RuleML-ISWC 2002 workshop) (2002)
3. Eades, P.: A heuristic for graph drawing. *Congressus Numerantium* 42, 149–160 (1984)
4. Fluit, C., Sabou, M., van Harmelen, F.: Supporting user tasks through visualisation of light-weight ontologies. In: Staab, S., Studer, R. (eds.) *Handbook on Ontologies*. Int. Handbooks on Information Systems, pp. 415–434. Springer, Heidelberg (2004)
5. Gabel, T., Sure, Y., Völker, J.: Kaon – ontology management infrastructure. SEKT informal deliverable 3.1.1.a, Institute AIFB, University of Karlsruhe (2004)
6. Gomez-Perez, A., Corcho, O., Fernandez-Lopez, M.: *Ontological Engineering*. Springer, New York (2003)
7. Gruber, T.R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. In: Guarino, N., Poli, R. (eds.) *Workshop on Formal Ontology*, Padua, Italy. *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers, Dordrecht (1993)
8. Guarino, N.: Understanding, Building, and Using Ontologies: A commentary to Using Explicit Ontologies in KBS Development. In: van Heijst, S., Wielinga (eds.) *International Journal of Human and Computer Studies* 46, 293–310 (1997), <http://citeseer.ist.psu.edu/guarino97understanding.html>
9. Halpin, T.A.: *Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design*. Morgan Kaufman Publishers, San Francisco (2001)
10. Halpin, T.A.: Business Rule Modality. In: Proc. Of Eleventh Workshop on Exploring Modeling Methods for Systems Analysis and Design, EMMSAD 2006 (2006), <http://www.orm.net/pdf/RuleModality.pdf>
11. Halpin, T.A., Curland, M.: Automated Verbalization for ORM 2. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops*. LNCS, vol. 4278, pp. 1181–1190. Springer, Heidelberg (2006)
12. Kripke, S.: Semantical Considerations on Modal Logic. *APF* 16, 83–94 (1963)
13. Lambrix, P., Edberg, A.: Evaluation of ontology merging tools in bioinformatics. In: *Pacific Symposium on Biocomputing*, pp. 589–600 (2003)
14. Meersman, R.: The use of lexicons and other computer-linguistic tools in semantics, design and cooperation of database systems. In: Zhang, Y., Rusinkiewicz, M., Kambayashi, Y. (eds.) *The Proceedings of the Second International Symposium on Cooperative Database Systems for Advanced Applications (CODAS 1999)*, pp. 1–14. Springer, Heidelberg (1999)
15. Mutton, P., Golbeck, J.: Visualization of semantic metadata and ontologies. In: *IV 2003: Proceedings of the Seventh Int. Conference on Information Visualization*, Washington, DC, USA, p. 300. IEEE Computer Society, Los Alamitos (2003)
16. Noy, N.F., McGuinness, D.L.: *Ontology development 101: A guide to creating your first ontology*. Technical Report KSL-01-05, Knowledge Systems Laboratory, Stanford University, Stanford, CA, 94305, USA (2001)
17. Pretorius, J.A.: Lexon visualization: Visualizing binary fact types in ontology bases. In: Lambrix, P., Edberg, A. (eds.) *IV 2004*, pp. 58–63 (2004); Evaluation of ontology merging tools in bioinformatics. In: *Pacific Symposium on Biocomputing*, pp. 589–600 (2003)

18. Spyns, P., Meersman, R., Jarrar, M.: Data modeling versus Ontology engineering. *SIGMOD Record: Special Issue on Semantic Web and Data Management* 31(4), 12–17 (2002)
19. Storey, M., Musen, M., Silva, J., Best, C., Ernst, N., Fergerson, R., Noy, N.: *Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in protégé* (2001)
20. Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., Wenke, D.: *OntoEdit: Collaborative ontology development for the Semantic Web*. In: Horrocks, I., Hendler, J. (eds.) *ISWC 2002*. LNCS, vol. 2342, pp. 221–235. Springer, Heidelberg (2002)
21. Tang, Y., Spyns, P., Meersman, R.: *Towards Semantically Grounded Decision Rules Using ORM+*. In: Paschke, A., Biletskiy, Y. (eds.) *RuleML 2007*. LNCS, vol. 4824, pp. 78–91. Springer, Heidelberg (2007)
22. Tang, Y., Meersman, R.: *On constructing semantic decision tables*. In: Wagner, R., Revell, N., Pernul, G. (eds.) *DEXA 2007*. LNCS, vol. 4653, pp. 34–44. Springer, Heidelberg (2007)
23. Trog, D., Vereecken, J., Christiaens, S., De Leenheer, P., Meersman, R.: *T-Lex: a Role-based Ontology Engineering Tool*. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops*. LNCS, vol. 4278, pp. 1191–1200. Springer, Heidelberg (2006)
24. Trog, D., Tang, Y., Meersman, R.: *Towards Ontological Commitments with O-RIDL Markup Language*. In: Paschke, A., Biletskiy, Y. (eds.) *RuleML 2007*. LNCS, vol. 4824, pp. 92–106. Springer, Heidelberg (2007)

DOGMA-MESS: A Tool for Fact-Oriented Collaborative Ontology Evolution*

Pieter De Leenheer and Christophe Debruyne

Semantics Technology and Applications Research Laboratory (STARLab)
Vrije Universiteit Brussel, Pleinlaan 2, Brussels 5, Belgium

Abstract. Ontologies being shared formal specifications of a domain, are an important lever for developing meaningful internet systems. However, the problem is not in what ontologies are, but how they become operationally relevant and sustainable over longer periods of time. Fact-oriented and layered approaches such as DOGMA have been successful in facilitating domain experts in representing and understanding semantically stable ontologies, while emphasising reusability and scalability. DOGMA-MESS, extending DOGMA, is a collaborative ontology evolution methodology that supports stakeholders in iteratively interpreting and modeling their common ontologies in their own terminology and context, and feeding back these results to the owning community. In this paper we extend DOGMA Studio with a set of collaborative ontology evolution support modules.

1 Introduction

Ontologies, being formal, computer-based specifications of shared conceptualisations of the worlds under discussion, are an important lever for developing meaningful communication between people and internet systems [9,8]. However, the problem is not in what ontologies are, but how they become *community-grounded* resources of semantics, and at the same time be made operationally relevant and sustainable over longer periods of time. The state of the art in ontology evolution regards change as a pain that must be technically alleviated by presuming a project-like practice where ontologies are created and deployed in discrete steps [5]. The requirements for the “ontology project” are usually deduced from the technical web service requirements that were solo-designed by a single application developer, rather than collaboratively grounding them directly in the community. In the DOGMA framework [12], *fact-oriented approaches* such as NIAM/ORM [20,10] have been proven useful for engineering ontologies. A key characteristic here is that the analysis of information is based on natural language facts. This brings the advantage that “layman” domain experts are facilitated in building, interpreting, and understanding attribute-free, hence semantically stable ontologies, using their own terminology. DOGMA-MESS is a teachable and repeatable *collaborative ontology evolution methodology* that supports stakeholders in interpreting and modeling

* We would like to thank Stijn Christiaens for his valuable comments on the usability of the tool. The research described in this paper was partially sponsored by the EC projects FP6 IST PROLIX (FP6-IST-027905) and FP7 TAS3.

their common ontologies in their own terminology and context, and feeding back these results to the owning community. In this paper we extend DOGMA Studio with a set of modules (Perspective Manager, Version Manager, and Community Manager) that support these fact-oriented collaborative ontology evolution processes.

2 DOGMA Ontology Engineering

The DOGMA¹ ontology approach and framework [14] is adopted with the intention to create flexible, reusable bounded semantics for very diverse computational needs in communities for an unlimited range of pragmatic purposes. DOGMA has some distinguishing characteristics that make it different from traditional approaches such as (i) its groundings in the linguistic representations of knowledge, (ii) the explicit separation of the conceptualisation (i.e., lexical representation of concepts and relationships) from its axiomatisation (i.e., semantic constraints) and (iii) its independence from a particular representation language. The goal of this separation, referred to as the *double articulation* principle [18], is to enhance the potential for reuse and design scalability.

Lexons are initially uninterpreted binary fact types, hence underspecified, which increases their potential for reusability across community perspectives or goals. Lexons are collected in a lexon base, a reusable pool of possible vocabularies, and represented as 5-tuples declaring either: a taxonomical relationship (*genus*): e.g., $\langle \gamma, \text{manager}, \text{is a}, \text{subsumes}, \text{person} \rangle$; or a non-taxonomical relationship (*differentia*): e.g., $\langle \gamma, \text{manager}, \text{directs}, \text{directed by}, \text{company} \rangle$, where γ is an abstract context identifier, lexically described by a string in some natural language, and is used to group lexons that are logically related to each other in the conceptualization of the domain.

Another distinguishing characteristic of DOGMA is the explicit *duality* (orthogonal to double articulation) in interpretation between the syntactic level and semantic level. The goal of this separation is primarily to disambiguate the syntactic representation of terms in a lexon into concept definitions, which are word senses taken from a *community glossary* such as WordNet². The meaning of the terms in a lexon is dependent on the *context of elicitation* [3]. For example, a term “capital” elicited from a typewriter manual (read: context γ), it has a different meaning (read: concept definition) than when elicited from a book on marketing. Though ontologies can differ in syntax, semantics, and pragmatics, they all are built on this shared vocabulary, called the lexon base.

The *perspective commitment layer* mediates between the lexon base and its applications. Each such perspective defines a partial semantic account of an intended conceptualization [9]. It consists of a finite set of axioms that specify which lexons of the lexon base are interpreted and how they are visible in the committing application, and (domain) rules that semantically constrain this interpretation. Experience shows that it is much harder to reach an agreement on domain rules than one on conceptualization [14]. E.g., the rule stating that each patient is a person who suffers from at least one disease may be too strong in some domains.

¹ Developing Ontology-Grounded Methods and Applications.

² <http://wordnet.princeton.edu/>

3 Community Evolution

Community dynamics, as illustrated in Fig. 1 is characterised by Nonaka’s [16] four modes of *knowledge conversion*: *socialisation*, *externalisation*, *combination*, and *internalisation*. At the heart of the community dynamics is the Ontology Server, that bridges the semiotic gap between the community system parts. It is embedded in a central ontology evolution support system we introduced [6] and validated [11] earlier. There are three types of *knowledge workers*: the *knowledge engineer*, the *core domain expert* (CDE), and the *domain expert* (DE). As we will show, in DOGMA-MESS, the involved ontology evolution processes (*community grounding*, *rendering*, *alignment*, and *commitment*) are inherently driven by the social knowledge conversion modes.

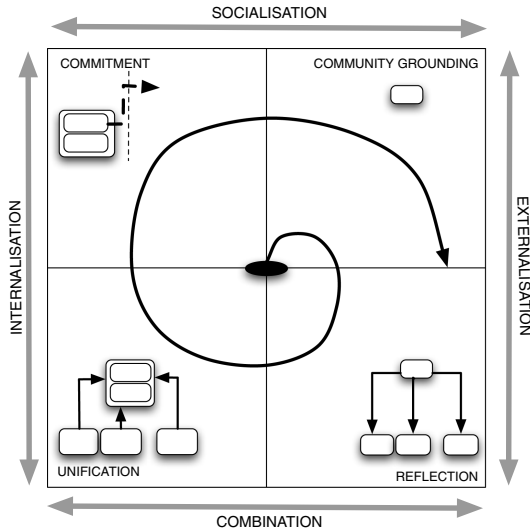


Fig. 1. DOGMA-MESS ontology evolution spiral model

1. **Community Grounding:** In this phase, shared conceptions of the world under discussion that emerged from *socialisation* are analysed by the CDE. With the assistance of the KE, he identifies the key *conceptual patterns* that are relevant to be further externalised to the ontology. This results in a generalised *upper common ontology* (UCO) which represents the conceptualizations that are common to and accepted by the community.
2. **Perspective Rendering:** All participating stakeholders’ DEs render their perspective on the UCO, by specialising the conceptual patterns, resulting in a set of diverging *stakeholder perspectives* (SPs). Doing so, ontology evolution is grounded (bottom-up) in the community, starting with the variety of terminologies found in the community itself. This allows DEs to syntactically and semantically nuance their intensions in a more natural manner using their own vocabulary. In order to impose UCO reuse, different types of *perspective reuse policies* can be formalised,

including articulation, specialisation, and application. A reuse policy is formalised by a set of applicable operations on a perspective (see [3]).

3. **Perspective Unification:** In the *lower common ontology* (LCO), a new proposal for the next version of the common ontology is produced, *combining* relevant material from the UCO and various stakeholder perspectives. Basically, there is only a very simple rule: all (selected) definitions need to be full specializations of the conceptual patterns in the UCO. This, however, is overly simplified. In the ontology evolution process, despite reuse policies, the constructivist paradigm should allow to *override* the reuse policies, and hence new definitions to be created that are not (complete) specializations, but represent new insights for the CDE in preparing new evolution rounds, for example. This makes the alignment process far from trivial. This process is conducted collaboratively by all involved DEs, the CDE, and the KE.
4. **Perspective Version Commitment:** The part of the LCO that is aligned by the community forms the legitimate UCO for the next version of the common ontology. All participating organisations finally internalise and commit their instance bases to the new version.

In all phases, the views of all stakeholders are considered. This fourfold collaborative ontology evolution process is iteratively applied until an optimal balance of differences and commonalities between organisational and common perspectives are reached that meets the communication goals.

4 DOGMA Studio

DOGMA Studio contains both a *Workbench* and a *Server*. The Workbench is constructed according to the Eclipse plugin architecture. The loose coupling allows any arbitrary community to support its own ontology engineering method by customised ontology viewing, querying or editing plugins. The Server is an advanced J2EE application running in a JBoss server which efficiently stores Lexons and Commitments in a PostgreSQL Database. The manual input method uses the NORM notation, which is an adaptation of NIAM/ORM2, introduced by [19]. Workbench can also perform conversion to and from the following formats: (i) comma-separated files can be imported in (exported from) the Perspective Base; (ii) Ω -RIDL commitment files can be imported in (exported from) the Perspective Commitment Layer; and (iii) RDF(S)/OWL files can be imported in (exported from) the Perspective Base [11]. Following is an overview of the three main Eclipse perspectives to support the community evolution processes, i.e. Version Manager, Community Manager, and Perspective Manager. Due to space limitations the overview is rather limited. For more demo material we refer to our website [5].

4.1 Version Manager

The Ontology Version manager provides basic plugins for Viewing and Editing Perspective and Pattern Versions.

³ <http://www.starlab.vub.ac.be/website/dogmastudio> (last access: 5 July 2008).

Ontology Viewer. plugin allows to explore the perspectives, patterns, perspective policies, and their versions that are currently stored in the Server. The tree directory view sorts the ontologies per domain, as shown in Fig. 2 (top). The first level enlists the domains (e.g., *opensourceartefacts*); the second level, within one domain, enlists the ontologies (e.g., *sodocu*); the third level, within one ontology, enlists the version history; the fourth level, within one ontology, enlists the available patterns (and their versions) (e.g., *software_artefact*); and finally, the fifth level, given a pattern, shows the perspective version history. To uniquely identify definition versions, we adopted a universe resource identifier (URI). E.g., *domain/ontology/pattern;j#stakeholder,i* identifies a perspective on a pattern, with version number j , rendered by stakeholder, with version number i .

Ontology Editor. plugin has three panes for viewing/editing the taxonomy, the relations, or the whole pattern or perspective that was selected from the Ontology Viewer. The latter is illustrated by Fig. 2 (second line) 4. The title bar shows the version URI. Each part of the perspective has a different colour: pattern parts are coloured blue, UCO parts are coloured grey, and perspective parts currently rendered by the DE are coloured in yellow. Figure 2 (third line) shows a fourth pane that displays the events and editing log, tracking each change operation made to the perspective. For each operation, the pre- and postconditions are shown indicating why the operation succeeded or failed. Change logs allows for semantic conflict analysis during *change-based merging* of parallel SPs. For a formalisation of this using *graph transformation theory* see e.g., [4]. Finally, the Concept Viewer plugin retrieves the concept definition from the Community Glossary when a term is selected.

4.2 Community Manager

The Community Manager provides plugins for managing *tickets* and conceptual patterns. During the community grounding phase, for each key concept of interest a ticket is created with attached a conceptual pattern, and sent to the relevant stakeholders in the community to render their perspective on it. The *Ticket Viewer* and the *Dialogue Box* are illustrated in Fig. 2 (bottom). A ticket has a title; informal information about the rendering task; a priority code (high, higher, normal, lower or low); the context in which the ticket is to be executed; the CDE who created the ticket; and finally, the stakeholding DEs receiving the ticket. A Dialogue Box is opened prompting the knowledge worker to open the ontology context in which the evolution task is to be performed. The *Ticket Maker* plugins are not illustrated due to space limitations.

4.3 Perspective Manager

The Perspective Manager provides a Conflict Viewer, a Conflict Browser, and finally, a Perspective Analyser.

⁴ The examples are in Dutch as they are extracted from a realistic case study (Sect. 4.4) that was conducted in the Netherlands.

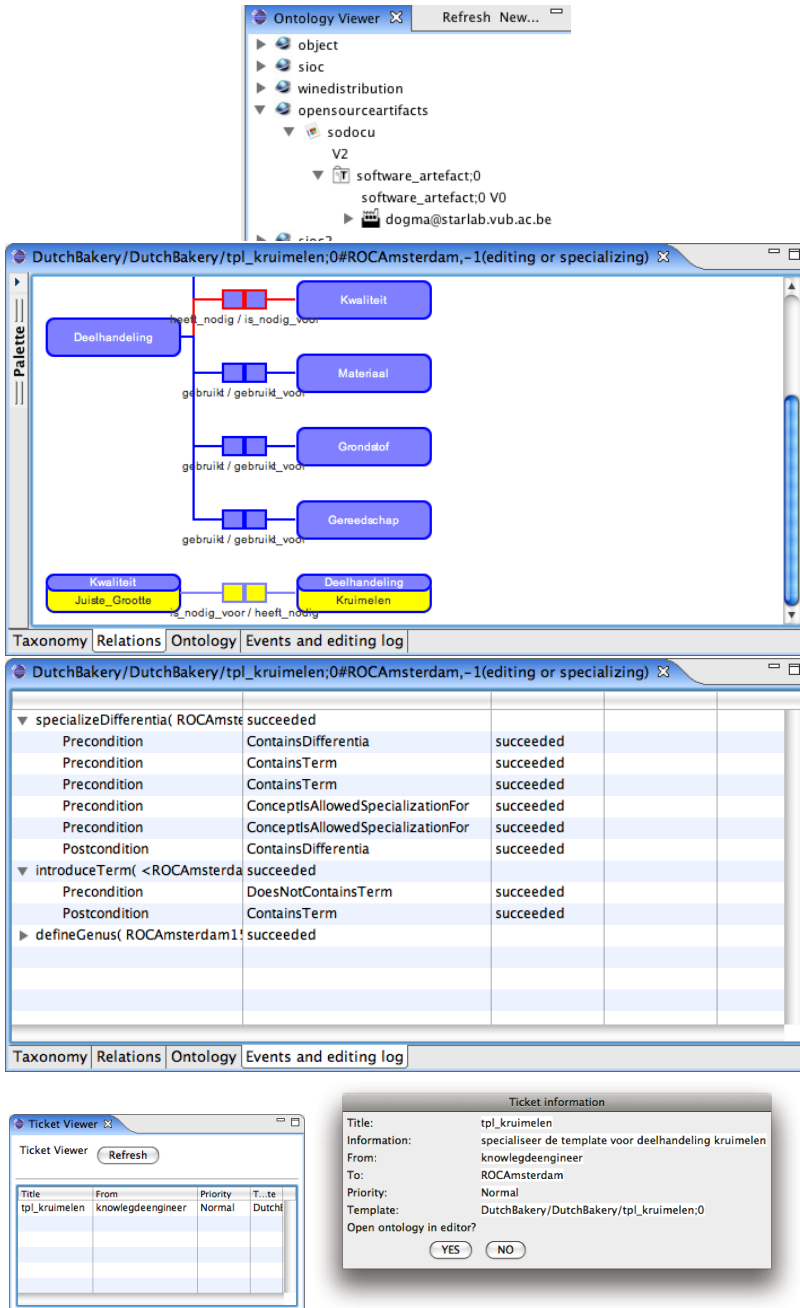


Fig. 2. From top to bottom: (1) the Ontology Viewer; (2) the integrated ontology viewing/editing pane of the Ontology Editor; (3) the events and editing log pane of the Ontology Editor; (4) the Ticket Viewer and its Dialogue Box

Conflict Viewer. plugin allows the exploration of conflicts against reuse policies in a graphical way. All conflicts have a conflict code that uniquely identifies the conflict type, and a conflict ID that facilitates retrieving more detailed information about the conflict in the Conflict Browser. From within the Conflict Viewer the stakeholder can check a perspective against all the different combinations of reuse policies that were defined in [3]. As a stakeholder can have multiple perspectives, he must first select the perspective he prefers to check.

Conflict Browser. Conflict Browser provides detailed information about all conflicts in a tree structure. The following table shows the meaning of three of the in total eleven defined conflict codes (*C_i*, 11 in total):

- C3: NewlyDefinedRelationConflict: a new relation was specified;
- C6: IntroduceTermConflict: a new term was introduced;
- C8: DefinedGenusWithNewConceptConflict: a genus was defined consisting of one or more newly introduced concepts.

When expanding the line, description of the involved concepts and relationships, and a cause for the conflict is revealed. This allows the DE in finding related conflicts that caused this particular conflict, and how it could be solved. Conflicts can be sorted based on conflict id, type, or cause. E.g., consider Fig. 3. The conflicts concern a perspective that has a specialisation reuse policy with a pattern in the UCO. When a specialisation policy holds, the DE is restricted to operators that specialise relationships by reusing concepts that were already defined in the UCO [3]. The first conflict with ID 6 has code

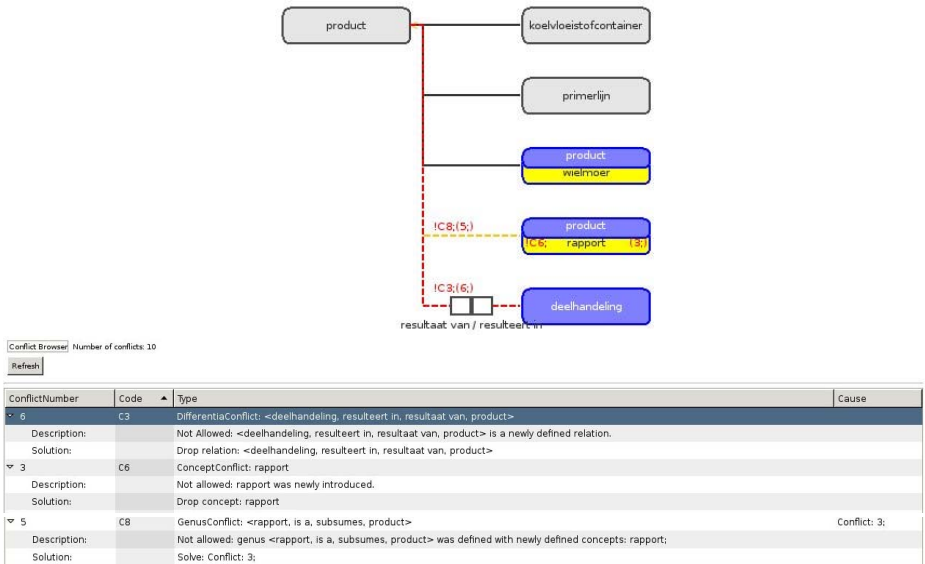


Fig. 3. The Perspective Manager Eclipse perspective

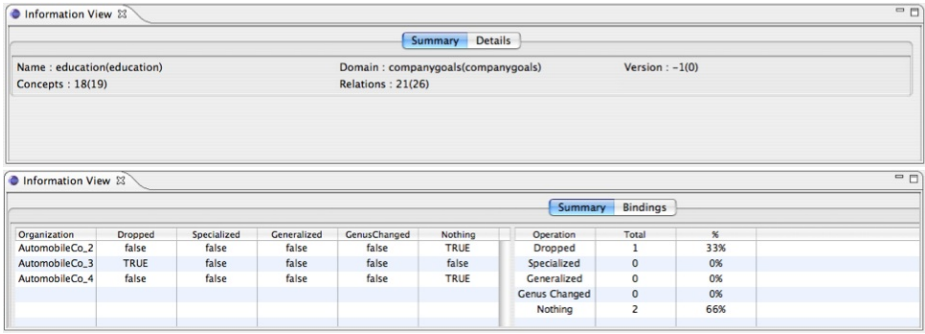


Fig. 4. Analysing Stakeholder Perspectives: the top view shows statistics when selecting an arbitrary definition. The bottom view is shown when selecting a term in a pattern.

C3, and indicates that a new relation is defined <Deelhandeling⁵, resulteert in, resultaat van, Product>. The proposed solution is to drop the relation. The second conflict with ID 3 and code C6 indicates that a term Rapport⁶ is introduced in the perspective while introducing new terms is restricted by the policy. The third conflict with ID 5 and code C8 was caused by conflict 6 as the introduction of <Rapport, is a, subsumes, Product> involves a newly introduced term Rapport, which is not allowed. The Conflict Browser also notes that conflict 5 would be automatically resolved if conflict 3 is resolved.

Perspective Analyser: plugin allows the Knowledge Worker to explore and analyse the differences and similarities between divergent stakeholder perspectives that were concurrently rendered on one original pattern. The plugin consists of two views, both illustrated in Fig. 4. When selecting a concept or relationship, summarising statistics are shown. When clicking on a term in a pattern, the Analyser parses for each stakeholder perspective the change log, and shows how the term evolved, illustrated on the bottom of Fig. 4.

4.4 Validation

We validated the DOGMA-MESS tools in the context of a realistic case study of the European CODRIVE⁷ project. This project aims at contributing to a competency-driven vocational education by using state-of-the-art ontology methodology and infrastructure in order to develop a conceptual, shared and formal KR of competence domains. Domain stakeholders included educational institutes and public employment organisations from various European countries. The resulting shared “Vocational Competency Ontology” will be used by all partners to build interoperable competency models. All the samples in this paper were drawn from this case study. For an elaboration on the case study and design choices made for DOGMA-MESS, we ref to [1] and [2].

⁵ Partial Activity.
⁶ Report.
⁷ CODRIVE is an EU Leonardo da Vinci Project (BE/04/B/F/PP-144.339).

5 Discussion and Future Work

We plan to extend the colour palette in the Ontology Editor. E.g., red could be used for parts that are deprecated. We also could use colour ranges to indicate the percentage of agreement on a certain concept in the UCO. The Community Manager will be further extended with an underlying community ontology that will give a semantic helicopter view on the current situation when creating tickets and patterns. First-class citizens of this ontology are inspired from related ontological frameworks. E.g., inspired by [7], a *community* is a special type of social system for which different directions and aims are set, as well-established common *goals* towards which the community strives in order to create *added value* and which normally are accomplished by coherent (collaborative) *actions* that are performed by subscribed legitimate *stakeholders* and where these actions are aiming at changing the community state in a desired way. Other examples include SIOC⁸. In order to support the perspective unification process, we are currently implementing a meaning negotiation and argumentation module, that is inspired by related tools such as HCOME [13] and Diligent [17]. Currently, we are implementing features to reuse facts not only at a lexical and a semantic level, but also at a pragmatic level, by extending the latter with an additional social layer to the semiotic fabric of DOGMA where domain experts can mark their favourite fact types or the actions on concepts are stored to distinguish the interesting concepts.

6 Conclusion

The problem in ontology engineering is not on what ontologies are, but how they become operationally relevant and sustainable over longer periods of time, and how proper methodology and tool support can be provided. DOGMA-MESS, extending the fact-oriented and layered ontology framework DOGMA, is a collaborative ontology evolution methodology that supports stakeholders in iteratively interpreting and modeling their common ontologies in their own terminology and context, and feeding back these results to the owning community. In this paper we extend DOGMA Studio with a set of MESS modules: Version Manager, Community Manager, and Perspective Manager.

References

1. Christiaens, S., De Leenheer, P., de Moor, A., Meersman, R.: Business use case: Ontologising competencies in an interorganisational setting. In: Hepp, M., De Leenheer, P., de Moor, A., Sure, Y. (eds.) *Ontology Management for the Semantic Web, Semantic Web Services, and Business Applications, from Semantic Web and Beyond: Computing for Human Experience*. Springer, Heidelberg (2008)
2. De Leenheer, P.: Meaningful competency-centric human resource management: a case study for Dogma Mess. In: *Proc. of European Semantic Technology Conference 2008, Vienna, Austria* (2008)
3. De Leenheer, P., de Moor, A., Meersman, R.: Context dependency management in ontology engineering: a formal approach. *LNCS Journal on Data Semantics* 8, 26–56 (2007)

⁸ <http://www.sioc-project.org>

4. De Leenheer, P., Mens, T.: Using graph transformation formal collaborative ontology evolution. In: Proc. of Agtive, Kassel, Germany. LNCS, vol. 5088. Springer, Heidelberg (2007)
5. De Leenheer, P., Mens, T.: Ontology evolution: State of the art and future directions. In: Hepp, M., De Leenheer, P., de Moor, A., Sure, Y. (eds.) *Ontology Management for the Semantic Web, Semantic Web Services, and Business Applications*. Springer, Heidelberg (2008)
6. de Moor, A., De Leenheer, P., Meersman, R.: DOGMA-MESS: A meaning evolution support system for interorganizational ontology engineering. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) *ICCS 2006*. LNCS (LNAI), vol. 4068, pp. 189–202. Springer, Heidelberg (2006)
7. Falkenberg, E.D.: FRISCO: A framework of information system concepts. Technical report, IFIP WG 8.1 Task Group (1998)
8. Gruber, T.R.: A translation approach to portable ontology specifications. *Knowledge Acquisition* 5(2), 199–220 (1993)
9. Guarino, N., Giaretta, P.: Ontologies and knowledge bases. towards a terminological clarification. In: Mars, N. (ed.) *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, pp. 25–32. IOS Press, Amsterdam (1995)
10. Halpin, T., Morgan, T.: *Information Modeling and Relational Databases*, 2nd edn. Morgan Kaufmann Publishers, San Francisco (2008)
11. Jarrar, M.: Mapping ORM into the SHOIN/OWL description logic. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM-WS 2007, Part I*. LNCS, vol. 4805, pp. 729–741. Springer, Heidelberg (2007)
12. Jarrar, M., Demey, J., Meersman, R.: On reusing conceptual data modeling for ontology engineering. *Journal on Data Semantics* 1(1), 185–207 (2003)
13. Kotis, K., Vouros, G.: Human-centered ontology engineering: The HCOME methodology. *Knowledge and Information Systems* 10, 109–131 (2005)
14. Meersman, R.: Semantic ontology tools in IS designs. In: Raś, Z.W., Skowron, A. (eds.) *ISMIS 1999*, vol. 1609, pp. 30–45. Springer, Heidelberg (1999)
15. Meersman, R., Tari, Z., Herrero, P. (eds.): *OTM-WS 2007, Part I*. LNCS, vol. 4805. Springer, Heidelberg (2007)
16. Nonaka, I., Takeuchi, H.: *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, Oxford (1995)
17. Pinto, H., Staab, S., Tempich, C.: Diligent: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In: *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)*, Valencia, Spain (2004)
18. Spyns, P., Meersman, R., Jarrar, M.: Data modelling versus ontology engineering. *SIGMOD Record* 31(4), 12–17 (2002)
19. Trog, D., Vereecken, J., Christiaens, S., De Leenheer, P., Meersman, R.: T-lex: A role-based ontology engineering tool. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops*. LNCS, vol. 4278, pp. 1191–1200. Springer, Heidelberg (2006)
20. Verheijen, G., Van Bekkum, J.: NIAM, an information analysis method. In: *Proc. of the IFIP TC-8 Conference on Comparative Review of Information System Methodologies (CRIS 1982)*. North-Holland, Amsterdam (1982)

Evaluation and Enhancements for NORMA: Student User Suggestions

Gordon C. Everest

Professor Emeritus
Carlson School of Management
University of Minnesota, USA
geverest@umn.edu

Abstract. In an advanced database design class, students used NORMA for the first time to do several data modeling assignments. One of the graded requirements was to write a comprehensive feedback memo directed primarily to the developers of the data modeling tool. The memo was to be broken down into good points, bad points, problems, and suggestions for improvement or enhancement. This provides a unique opportunity to gather substantial feedback from users, since the students are highly motivated to use the tool to complete each assignment, working through any problems or frustrations they faced. Giving up was not an option.

This paper provides a summary report of the experiences and suggestions as captured in the feedback memos. Several suggestions would be relevant for other fact-oriented design tools. Overall, students really liked using the tool, finding it to be reasonably intuitive and easy to use. Data model verbalization was the most frequently mentioned benefit. The major problems related to a lack of good documentation, the convoluted and difficult process of generating a database (DDL script), and the lack of adequate data model abstraction mechanisms.

Keywords: data modeling, database design, database design tools, data modeling tools, CASE tools, Object Role Modeling (ORM), NORMA.

1 Background and Introduction

The Course and the Students. Every spring, I teach a 15-week course in Advanced Database Design. In 2008, two sections had a total of 21 students – 17 full time MIS majors (most seniors; many working part-time or as an intern in IT), and 4 professionals - two as data modelers. Students were given four progressively more difficult design assignments using NORMA. They self-reported spending an average of 20 hours total on these assignments – the longest time typically five times the shortest time.

The course first reviews Entity-Relationship (ER) and Relational modeling, and normalization. The students are expected to have a reasonably good grasp of these ideas from prerequisite courses or prior experience. Then the last (more than) half of the course is spent on Object Role Modeling exclusively.

To facilitate learning ORM, students use the data modeling tool NORMA [3].

Usage Notes for NORMA. In addition to the documentation available on the open source website and the Halpin text [2], the instructor prepared some *Usage Notes* [1] – screen shots are avoided to keep the file size and page count small. The notes are the opposite extreme from the Lab notes – by design, more like a cryptic crib sheet – intended to be referenced frequently as one uses the system. They succinctly show step by step how to accomplish most tasks in NORMA. Being the first year, the *Notes* were initially prepared solely on the basis of the instructor’s limited usage experience.

Feedback Memo. As part of their course grade, students write a feedback memo primarily for the modeling tool vendor/developer. Based on their experience using the design tool for their assignments, the memo includes how the system or its documentation can be improved. The memo assignment is as follows:

In a feedback memo, please add anything to the *Usage Notes* which you feel would be helpful to future students or the vendor. Begin to prepare your memo as a compilation of all the comments you had written in earlier assignments. In your memo, separate out your comments into the following categories:

1. Good points, compliments
2. Bad points, frustrations
3. Problems, work arounds, how to’s, and helpful hints for using the current system; anything which should become additions to the *Usage Notes*.
4. Suggestions for improvement – added capabilities and functionality; modifications to the user interface to make the system easier to use and more intuitive.

No specific length. It will be scored on substance.

Beyond the suggestions above, the student responses were completely unstructured. Whatever they included in their memos was self-selected, and not directed by any specific list of questions, topics, or features. Consequently, any metrics on the frequency with which any particular compliment, problem, or suggestion was mentioned would not be very meaningful. Tallying responses to specific questions or features may be a good idea in the future to obtain more consistent and reliable results. Also, the student feedback memos were necessarily influenced by the instructor – what was presented in class, and the *Usage Notes*.

The students collectively had many great insights, ideas, and suggestions. Students are doubly motivated to think critically and creatively when using a system – they have to reach a point of completion for each assignment. In addition, the feedback memo contributes directly to their grade! Students are encouraged to reflect on their experiences and to make constructive suggestions. This provides a unique opportunity for the developer/vendor to improve the product.

This paper focuses mainly on student’s overall evaluation, and on suggestions for improvement and enhancement, rather than on the obvious bugs that need fixing. No attempt has been made to rank or tally the frequency of items mentioned. However, most mentioned in this paper appeared multiple times. Some student comments or suggestions are augmented by the author to provide clarification, elaboration, and some additional suggestions based upon his own experience or lectures.

Parts of this paper will be of interest for data modeling tools in general, not just NORMA. Other parts will be of interest to both users and developers of NORMA. For this paper, the students were providing feedback on a version of NORMA from about

2008 March (running under Visual Studio 2005). Some problems may have subsequently been fixed, or enhancements already made. To someone already familiar with NORMA, some of the comments and suggestions are familiar and expected.

If ORM is to make inroads into data modeling practice around the world (and it should), then we must have a supporting software design tool which is easy to use, intuitive for the novice and efficient for the experienced user, industrial strength to support what enterprise data modelers need to do, and well documented with a website to get answers to questions, solutions to problems, and to submit suggestions for improvement. Hopefully, this paper will begin/continue a serious dialogue on what an ORM design tool should do, how, and in what priority.

Even though students were asked to separately discuss problems and suggestions, they have been combined in this paper. Following the next section of good points and compliments, the bulk of the paper covers the many suggestions for improvement and enhancement which would be of interest to developers, several stemming from particular problems or frustrations.

2 Good Points – Compliments

Nearly all the students commented on how much they liked and enjoyed using NORMA. Several said they would use it in their own current data modeling projects at work. Some said they would try to get it introduced in their organization.

Relatively independent of Visual Studio – that is, a user does not require a lot of knowledge about VS to use NORMA. At the same time, it is well integrated, using features similar to one who already knows Visual Studio.

User Interface – is intuitive and easy to use... once you learn how and get used to it. While several students praised NORMA for its user interface, others note some difficulties and inconsistencies (covered in the next section).

Verbalization – The benefit of the verbalizer was mentioned the most frequently. Some said it was the most helpful feature by far, the best part of using NORMA. It is very helpful in building a correct model. Additions or changes to the model diagram are instantly reflected in the verbalization – the designer can immediately verify that the diagram says what was intended.

Relational Table View – is very helpful in building a correct ORM model. Since most data modelers understand table-based (relational) data structures, this can serve as a check on the ORM model if the tables don't come out as expected. This also helps people to learn ORM when they can see the equivalent representation in tables.

Fact Editor – removes the busy work from placing construct icons in a diagram.

Sample Populations – using real sample data which is more meaningful to the designers and users, and can be used as test data in the operation of the database.

Optional display of Fork notation – gave a more visually intuitive representation of multiplicity in a relationship. Most students turned this feature on (of course, they were encouraged by the instructor!)

3 Problems, Suggested Improvements and Enhancements

These are organized by topical areas – system features, functions, and activities. When discussing a problem, it is useful to immediately follow it with some ideas for solution; when discussing limitations, it is useful to couple that with some suggestions for improvement or enhancement with added features. The suggestions may apply specifically to NORMA or may have broader implications for other data modeling tools. Some of the suggested improvements would take a minimum of effort to implement. Other suggestions represent a major work effort, but may have a large payoff for the success of NORMA in the business world of data modeling professionals.

3.1 User Interface

Guidelines for Consistent, Intuitive Interface. Many of the problems encountered by users begged for better documentation. However, most users don't or won't read the documentation. Even if they try initially, they soon give up with NORMA. Thus, it is important to have a consistent and intuitive user interface. There are many examples reported by the students which give evidence that the user interface needs improvement.

1.1 *Suggestion:* Do a thorough review of creation, display, editing, and deletion actions on different model elements to ensure that they are done in a consistent way.

1.2 *Suggestion:* Document the guidelines under which navigation, mouse clicking, menus, etc. were designed to function. If such guidelines do not exist, then develop explicit standards to guide developers in the design of the user interface, and how a user performs similar or related functions.

Managing Tool Bars and Windows. The system offers a lot of functionality and flexibility for controlling the placement and hide/show of toolbars. However, it is very confusing and daunting for the beginner. It is particularly frustrating figuring out how to get back a window or tool bar that disappeared (inadvertently or by choice).

1.3 *Suggestion:* Document the options behind the movement and placement of tool bars and windows in the user interface.

Drop Down Menus. Most of the drop-down menu choices do not apply to NORMA.

1.4 *Suggestion:* Grey out the menu choices which do not apply to NORMA, if it is to continue as a plug-in to Visual Studio.

3.2 Documentation and Help

Several student comments and suggestions concerned the availability and quality of documentation and help information. Several students commented about the help system, such as: "Gave up using Help" or telling future students, "Don't use the system Help." It gives too much noise with information that does not relate to NORMA. Clicking a hot link takes you to an online website where the information given was unrelated to NORMA.

2.1 *Suggestion:* This argues for decoupling NORMA from Visual Studio, or at least having a context sensitive link.

2.2 *Suggestion*: Every software system needs to have several types of documentation if it is to be successfully used:

- Release Notes – what has been fixed; known problems and workarounds; what is planned to be fixed or added, and in what priority.
- Tutorial and hands on exercises – somewhat fulfilled by the NORMA Lab notes.
- User Reference manual – with architectural overview of the system; organized to enable lookup of a specific topic either in the table of contents or the index.

All of these could potentially be available through the system online help and be separately available on the open source website, as are the Lab notes and Technical papers. The real need is for a place to go for answers to specific questions or on particular features. This would likely be in a reference manual, which could be online in help.

While the Lab notes are helpful for getting started, some other sort of reference documentation is necessary to lookup specific topics or problems when they arise. Looking something up when seeking the answer to a specific question is hard because the lab notes are difficult to navigate.

2.3 *Suggestion*: Even just adding a table of contents (and/or an index) to the numbered slides would make the Lab notes more useful.

2.4 *Suggestion*: More examples. They should be in the online help. For example, one is needed showing how to define a paired (composite) role set constraint.

2.5 *Suggestion*: Explain that selecting a reference mode also selects a default data type, which the designer should check and may want to change in the Properties window.

3.3 Operating Environment and Error Handling

Installation of Visual Studio Required First is a significant barrier to using NORMA. Many potential users and organizations do not have it installed. To many, the cost, time to install, and the disk space required represent significant hurdles.

While desirable for ORM to be (come) an integral part of an application system development environment such as VS.net, it should not be at the expense of users just desiring to do data modeling in ORM. Perhaps VS.net provides a useful platform for the development of NORMA, easing the work of the developer. However, VS.net provides little value added for the ORM data modeler.

3.1 *Suggestion*: It would be highly desirable to decouple NORMA from Visual Studio. It is confusing to be presented with all the options pertaining to Visual Studio, which have nothing to do with NORMA. The *Usage Notes* were helpful in this regard – telling us where to look for stuff that pertained to NORMA, and what to ignore.

3.2 *Suggestion*: Double or right click on an .ORM file in Windows to start up NORMA.

Help users Prevent and Correct Errors.

3.3 *Suggestion*: Start by telling the user what is always needed. For example, every entity object type (not value object types) must have a reference mode, except, it is optional for subtypes. Also, every predicate must have at least one reading, and a uniqueness constraint (except a unary predicate where there is only one possibility).

3.4 *Suggestion*: For each error, besides showing where it occurs in the diagram or in the Properties window, provide an explanation and possible corrective actions. The user should be able to jump to help for any given error and obtain a short tutorial and examples of how to do it right. Take every error as a teaching moment; a beginning user will want to go there, while an experienced user need not.

3.4 Model Construction and Manipulation

Several students mentioned the frustration of deleting an object, only to see it reappear in the object window pane or in the table diagram. Perhaps this is because they did not understand the difference between removing it from the diagram only, and deleting it from the underlying repository. The dialogue surrounding deletion of elements from the diagram (only or also from underlying model) is confusing enough that several students missed it.

4.1 *Suggestion*: At a minimum, the dialogue needs to be clarified. Perhaps make the default be delete from the model entirely, with a prompt to confirm or just want to remove it from the current diagram. An 'undo' operation would also be useful. Most of the time, when students deleted stuff from the diagram, they really intended it to be gone completely.

4.2 *Suggestion*: Add a reference mode for date and/or time to correspond to the temporal data types. Some possible formats: YYYY, YYMM, YYDDD, YYMMDD (with separators), YYYYMMDD, YYYY month DD, etc. all of which conform to the ISO and ANSI standards adopted in 1969 putting the year first.

4.3 *Suggestion*: Show all the relevant properties in the properties window, e.g., uniqueness and mandatory as well as readings on a predicate; allow changes from there.

4.4 *Suggestion*: Explain that readings can be deleted using only the Readings Editor, and that stray readings remain when you delete a fact type.

4.5 *Suggestion*: Fix adding a value object type from the tool box to a diagram (either by drag and drop, or select and place). Otherwise, explain that it is necessary to first add an entity object type then change its value type property to 'true.'

3.5 Constraints

The inconsistent treatment of constraints makes the system difficult to use. The only way to change some constraints is to delete and recreate. Uniqueness constraints do not appear where expected in the Properties window. To create a role constraint (uniqueness or mandatory) you right click on the role box; but to delete you must click on the constraint itself. It is not obvious how to delete/change a uniqueness constraint.

5.1 *Suggestion*: Constraints should be added, changed, and removed in the same way.

5.2 *Suggestion*: It would be useful to have a properties window for constraints, just as for objects. Then you could select or open a constraint, see all of its properties displayed, and be able to change them. It would also be nice to be able to change a constraint within the same family of constraints, as for example, to change a subset role constraint to an equality constraint.

When trying to define a frequency constraint on a unary predicate, the system converts it to binary and then does not allow the frequency to be set less than two.

It was difficult to select the correct role in a predicate, and to select them in the correct sequence when defining some of the constraints.

5.3 *Suggestion*: Enter a range of values as n-m, the industry standard (instead of n..m).

5.4 *Suggestion*: Add a frequency constraint on an object population (as with role).

3.6 Exporting and Copying Diagrams

Several students never discovered ‘Copy Image’ or knew what it meant. Selecting all or part of the diagram and typing CTRL-C did not work. Some students resorted to capturing screen shots to migrate the diagram to other applications.

When a ‘copied image’ was pasted into PowerPoint, it displayed OK but upon printing, big blobs appeared. One student discovered the cause: ungroup the diagram and you will find some small line segments are defined with a line width of ~100 points! Find and select those elements and redefine with a line width of 1 or 2 points.

3.7 Sample Population Data

Inputting sample data is tedious – the user must point and click with a mouse to move to the next value box or next line, thus moving back and forth from mouse to keyboard.

7.1 *Suggestion*: Clean up the user interface when entering sample data. Allow use of the TAB or ENTER key to move to the next input field or line.

When some objects already have sample values stored, the system does not always pick up those values when the object is used in a different context, such as in a ternary relationship (like an attribute being added to a relationship), in a subtype, or in an objectified predicate.

7.2 *Suggestion*: Whenever an object is involved in some relationship, once a sample population is given for a predicate, those values ascribed to an object in that relationship should be available when defining sample populations for other predicates, particularly with subtypes.

7.3 *Suggestion*: Allow input of sample values for an object by itself, then can use that data in the drop down box when entering sample population data for a predicate.

7.4 *Suggestion*: Enable importing/exporting from/to tables in Excel, Word, or Access.

3.8 Verbalization

Some verbalizations are quite convoluted and not easy for the user to understand. For example, with “at most one” it is easy to miss that it really means “zero or one.” The missing verbalizations of frequency and ring constraints is a serious limitation, and should receive high priority for implementation.

With a ring fact type having the same object playing three roles the verbalization was “Any object, object, object combination can occur only once.”

8.1 *Suggestion*: add “... in that particular order.”

8.2 *Suggestion*: When a reflexive relationship is defined, the system should prompt the designer to consider applying ring constraints. At least verbalize that they can appear in any order and that any pair may be the same instance.

3.9 Reports

Currently, NORMA generates two HTML reports: a `ConstraintValidationReport` and an `ObjectTypeList` (which is just that, a list). In addition, there are underlying files for each fact type and each object type. Clicking on an object or a predicate hotlink in these reports will bring up the detail.

9.1 *Suggestion*: Explain how the user can printout a single report with all the relevant or desired model information for all objects, fact types, constraints, notes, sample values, physical data types, etc.

9.2 *Suggestion*: Provide the ability to individually select the type of model elements to include in an output report, thus enabling the generation of somewhat tailored reports.

9.3 *Suggestion*: Offer a similar report on a generated relational table view.

3.10 Diagram Presentation – Abstractions

NORMA currently offers little in the way of enhancing a diagram to make it more readable and give additional semantic information about the model. Partitioning a large data model into multiple pages, and the context window are the abstraction mechanisms currently available in NORMA. Considerably more can be done to enable the ORM diagrams of a data model to be presented in a way that facilitates human understanding.

The biggest problem in presenting data model diagrams to people is handling complexity in large data models. The strategies for dealing with bigness and complexity can be categorized as follows:

1. Differentiation, Encoding, Layout
2. Abstraction
 - a. Scope – partitioning (within a boundary or ‘fence’)
 - b. Focus – displaying local detail in its global context (around a ‘point’)
 - c. Depth – suppressing detail; looking at less
3. Navigation over a model diagram – windowing, scrolling, panning, zooming

3.10.1 Differentiation

As a general principle, things that are different should look different. The corollary being: differences which carry no semantic significance should be avoided. It is noise to the reader if differences are not meaningful, e.g., colors or shapes.

10.1 *Suggestion*: In order for the designer to reflect the level of semantic importance for elements in a model diagram, provide the ability to vary the icon size, shape, line style (weight, and type, e.g., dotted), color (fill, line, or font), and font size and style. Allow the use of pictures or other graphical representations of objects which would make it easier for people to recognize and remember the object type.

3.10.2 Layout and Notes

More can be done to help the designer produce nice looking, organized diagrams.

10.2 *Suggestion*: When reversing roles, resolve the crossing lines and the direction of the reading at the same time. The user can always override.

10.3 *Suggestion*: Give a word wrap option when entering model notes. It’s not obvious that you must type SHIFT-ENTER to start a new line so at least document that somewhere.

10.4 *Suggestion*: Allow a model notes box to be with or without a border.

10.5 *Suggestion*: Allow a model notes box to be attached (with a dotted line) to a diagram element (or object). If the object is moved the box stays attached.

3.10.3 Abstractions

Abstraction means ‘leaving something out,’ so an abstraction *hides* things in the underlying model when presenting it to people. In NORMA, the diagram for a model can span multiple pages. Normally they would be partitions of a large model. Model elements may appear multiple times on the same or different pages, but there is only one representation in the underlying repository.

10.6 *Suggestion*: Make it possible to store various abstractions on separate pages.

Focus is highlighting a single selected element, and displaying only the model elements directly connected to it. The user selects how far out to reach from the selected object – a path (‘levels’ in NORMA) of one, two, or three segments (‘generations’) away. NORMA provides this abstraction mechanism in the Context Window – currently transitory and read only.

10.7 *Suggestion*: Make it possible to save abstractions in additional pages of the overall model diagrams. As it is currently, the objects are treated as copies from one page to another. It should also be possible to edit from an abstract view. Since all the information is maintained in a single underlying repository, any changes would be reflected in all instances in all the model diagrams, including all the abstractions. It is possible to simulate the focus abstraction (context window) on a separate page by dragging and dropping the desired objects connected to a focus object.

Depth - Levels of Detail. This is perhaps the most difficult abstraction mechanism to implement but may be the most important.

10.8 *Suggestion*: Enable the modeler to prepare a series of depth abstractions, each successively removing detail from the diagram. These can then serve as a way to present the model to people in reverse order starting with the highest level of abstraction, and progressively unfolding with increasing detail through a series of diagrams. Elements of an ORM diagram could be removed from the least important to the most important. The following list is a possible ranking:

1. Value object types and Lexical Object Types (LOTs)
2. ‘Terminal’ objects, playing only functionally dependent roles on other objects
3. Common objects (generic value domains or reference modes)
4. ‘Event’ objects
5. Dependent (‘weak’) objects, subtypes, objectified predicates
6. User-defined priority levels on object types
7. Constraints and reference modes
8. Predicate boxes

The end result would be like an ER diagram showing only base (independent) entities and relationship arcs. Ideally, the user should be able to specify what to exclude and what to include when producing a depth abstraction. Each abstraction diagram is then stored as another page of the model.

3.10.4 Simplifying an ORM Diagram

10.9 *Suggestion:* Allow a ‘terminal’ object to be displayed with no outline and suppress its predicate boxes (unless there are some additional semantics or constraints on the predicate). Terminal objects become attributes, and not an entity table, in the relational view. This is a special case of depth abstraction.

3.11 Relational Table View

Sometimes the default column names are very convoluted and the default order of the columns is undesirable. Currently, the primary keys and foreign keys get moved around with no apparent consistency.

11.1 *Suggestion:* Prompt the designer to rename column headings, particularly for foreign keys and objectified predicates.

11.2 *Suggestion:* Changed column names in the table diagram should be migrated back to and maintained within the ORM model, so that when you revise the ORM model and rebuild the tables, the changed names are retained.

11.3 *Suggestion:* Allow the designer to reorder the columns in a table and have this information retained in the repository.

11.4 *Suggestion:* Verbalize the relational table diagram.

11.5 *Suggestion:* Optionally display the properties of individual columns.

11.6 *Suggestion:* Allow model notes to be added to a table diagram (as for ORM).

3.12 Database Generation

Having to define a project before generating the DDL was very confusing and unnecessary. It should be just as easy as generating the relational table view.

12.1 *Suggestion:* Add a tabbed window in the diagram/document area for generic ANSI SQL. This avoids the need to set up a project in order to see the formal definition of the database in SQL. When the user is ready to generate the database definition for some specific target DBMS, that may be the appropriate time to require setting up a project folder within which to keep all the diagrams and definitions for the designed database.

4 Summary and Conclusions

NORMA was deemed above the threshold of acceptability and hence suitable for student use this year for the first time. However, the reported incidence of errors, bugs, and crashes, etc. would suggest that it is still not ready for enterprise data modeling in the corporate world. At the same time, NORMA represents a solid base and could have a major impact on the practice of data modeling if it is strengthened according to many of the suggestions in this paper.

We leave it for the NORMA developers and the ORM community to set priorities on implementing the suggested improvements and enhancements, determining how much effort is involved, and designing how to best implement the suggestions.

References

1. Everest, G.C.: Usage Notes for NORMA. University of Minnesota, Carlson School of Management, 2008 March 27, 6 pages. Includes genesis of ORM modeling tools; references to documentation; installing and starting NORMA; overview, architecture, and general user interface; navigating the diagram/document window; creating an ORM diagram; predicates, readings, and role names; adding constraints, ring fact types, and subtypes and associated constraints; formatting, displaying, and verbalizing an ORM diagram; naming and saving an ORM diagram; generating a relational model view (table diagram); generating the schema DDL script; printing and obtaining output; abstractions for presentation; and finally, obtaining help within NORMA. A copy of the updated April version, <http://www.ormfoundation.org/files/folders/norma/default.aspx>
2. Halpin, T.A.: Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design, Using ORM with ER and UML. Morgan Kaufmann, San Francisco (2003)
3. Neumont ORM Architect (NORMA). Open source software, <http://sourceforge.net/projects/ORM>

PerSys 2008 PC Co-chairs' Message

We welcome our workshop attendees and the readers of the PerSys 2008 proceedings to a very interesting collection of presentations and discussions. We received many good papers this year and had to limit the acceptance rate to 50%.

The papers cover a range of topics from middleware, context aware services, pervasive games, security, and privacy, to wireless and sensor networks showing the scope of interests and activities represented by the PerSys participants. PerSys again held its trademark discussion session, where all participants were able to present and discuss their research work, new ideas and directions.

We would like to thank everyone who submitted a paper, and we believe you will all have found the presentations and discussions stimulating and worthwhile. The reviewers of the papers did a terrific job in helping authors prepare better final papers and deserve special thanks for completing the job promptly.

Finally, we would like to sincerely thank the organizers of the conference for the hard work they put in to providing us with the opportunity to gather, present, listen, and discuss our research.

November 2008

Roy Campbell
Skevos Evripidou
Anja Schanzenberger

Learning, Prediction and Mediation of Context Uncertainty in Smart Pervasive Environments

Sajal K. Das and Nirmalya Roy

Center for Research in Wireless Mobility and Networking (CRWMan)
Department of Computer Science and Engineering
The University of Texas at Arlington
Arlington, Texas, 76019, USA
{das,nroy}@uta.edu

Abstract. The essence of pervasive computing lies in the creation of smart environments saturated with computing and communication capabilities, yet gracefully integrated with human users (inhabitants). *Context Awareness* is the most salient feature in such an intelligent computing paradigm. Examples of contexts include user mobility and activity among others. This paper reviews our work towards managing context uncertainty in smart pervasive environments. First we discuss a novel game theoretic learning and prediction framework that attempts to minimize the joint location uncertainty of inhabitants in multi-inhabitant smart homes. Next we present an ambiguous context mediation framework for smart home health care application. Finally, we describe an efficient, quality-of-inference aware context determination framework in pervasive care environments. We also present open problems in this area.

Keywords: Multi-inhabitant smart homes, context-aware computing, location/activity prediction, sensor data streams, smart health care.

1 Introduction

Advances in smart devices, mobile wireless communications, sensor networks, pervasive computing, machine learning, middleware and agent technologies, and human computer interfaces have made the dream of smart pervasive environments a reality. According to Cook and Das [3], a “smart environment” is one that is able to autonomously acquire and apply knowledge about its users (or inhabitants) and their surroundings, and adapt to the users’ behavior or preferences with the ultimate goal to improve their experience in that environment. The type of experience that individuals expect from an environment varies with the individual and the type of environment considered. This may include the safety of users, reduction of cost of maintaining the environment, optimization of resources (e.g., energy bills or communication bandwidth), task automation or the promotion of an intelligent independent living environment for health care services and wellness management. An important characteristic of such an intelligent, pervasive computing and communication paradigm lies in the autonomous

and pro-active interaction of smart devices used for determining users' important contexts such as current and near-future locations, activities, or vital signs. In this sense, *context awareness* is a key issue for enhancing users' living experience during their daily interaction with smart devices including sensors and computer systems, as only a dynamic adaptation to the task at hand will make the computing environments just user friendly and supportive.

Context awareness is concerned with the situation a *device* or *user* is in, and with adapting applications to the current situation. But knowing the current context an application or system is used to and dynamically adapting to it only allows to construct *reactive* systems, which run after changes in their environment. To maximize usefulness and user support, systems should rather adapt in advance to a new situation and be prepared before they are actually used. This demands the development of *proactive* systems, i.e., systems which predict changes in their environment and act in advance.

To this end, we strive to develop methods to learn and predict future contexts as well as to mediate ambiguous context, thus enabling systems to become proactive. Our goal is to provide applications not only with information about the current user contexts, but also with predictions of future contexts. When equipped with various sensors, a system should classify current situations and, based on those classes, learn the user behaviors and habits by deriving knowledge from historical data. The focus of our work is to forecast future user contexts lucidly by extrapolating the past and derive techniques that enable context prediction in pervasive systems. An instance of such an intelligent indoor environment is a *smart home* [4] that perceives the surroundings through sensors and acts on it with the help of actuators. In this environment, the most important contexts like user mobility and activity create an uncertainty of their locations and hence subsequent activities. In order to be cognizant of such contexts, the smart environment needs to minimize this uncertainty.

1.1 Contributions

In this paper we first summarize (in Section 2) an information-theoretic learning and prediction framework that minimizes context uncertainty in multi-inhabitant smart homes. This framework is based on Reinforcement Learning (Cooperative) [8] [9] and Nash Q -learning (Non-cooperative) [10] algorithms. The novelty of our work lies in the development of learning algorithms that exploit the correlation of mobility patterns across multiple inhabitants and attempts to minimize their joint uncertainty. This is achieved with the help of a *joint utility function* of entropy. Optimization of this utility function asymptotically converges to *Nash Equilibrium* [6]. Minimizing this utility function of uncertainty helps in accurate learning and estimation of inhabitants' contexts (locations and associated activities). Thus, the system can control the operation of automated devices in an adaptive manner, thereby developing an amicable environment inside the home and providing sufficient comfort to the inhabitants. This also aids in context-aware resource management, for example, minimizing the energy usage and hence reduction of overall maintenance cost of the house.

Next we envision sensor rich computing and networking environments that can capture various types of contexts of patients (or inhabitants of the environment), such as their location, activities and vital signs with applications to smart home health care. Such context information is useful in providing health related and wellness management services in an intelligent way so as to promote independent living. However, in reality, both sensed and interpreted contexts may often be ambiguous or uncertain, leading to fatal decisions if not properly handled. Thus, a significant challenge facing the development of realistic and deployable context-aware services for health care applications is the ability to deal with ambiguous contexts to prevent hazardous situations. In Section 3 of this paper, we summarize a quality assured context mediation framework, based on efficient context-aware data fusion and information theoretic system parameter selection for optimal state estimation in resource constrained sensor networks [11]. The proposed framework provides a systematic approach based on dynamic Bayesian network to derive context fragments and deal with context ambiguity or error in a probabilistic manner. It has the ability to incorporate context representation according to the applications' quality requirement [13].

Finally, we claim an energy-efficient determination of an individual's context (both physiological and activity) is also an important technical challenge for this assisted living environments. Given the expected availability of multiple sensors, context determination may be viewed as an estimation problem over multiple sensor data streams. In Section 4 we give an overview of a formal and practically applicable model to capture the tradeoff between the accuracy of context estimation (or determination) and the communication overheads of sensing [12]. In particular, we propose the use of *tolerance ranges* to reduce an individual sensor's reporting frequency, while ensuring acceptable accuracy of the derived context. Thus this framework is quality-of-interference (QoINF) aware. Experimental results with SunSPOT [15] sensors are also presented to attest to the promise of this approach.

We conclude this paper with open directions for research. (Details of our approaches are presented in [14].)

2 Mobility/Activity Learning and Prediction

From information theoretic view point, an inhabitant's mobility and activity create an uncertainty of their locations and hence subsequent activities. In order to be cognizant of such contexts, a smart pervasive environment needs to minimize this uncertainty as captured by Shannon's entropy measure [2]. An analysis of the inhabitant's daily routine and life style reveals that there exist some well defined patterns. Although these patterns may change over time, they are not too frequent or random, and can thus be learnt. This simple observation leads us to assume that the inhabitant's mobility or activity follows a *piece-wise stationary, ergodic stochastic process* with an associated uncertainty (entropy), as originally formulated in [1] for optimally tracking (estimating and predicting) location of mobile users in wireless cellular networks. In [7], the authors adopted

the framework from [1] to design an optimal algorithm for location (activity) tracking in a smart environment, based on compressed dictionary management and online learning of the inhabitant’s mobility profile, followed by a predictive resource management (energy consumption) scheme for a single inhabitant smart space. However, the presence of multiple inhabitants with dynamically varying profiles and preferences (selfish or non-selfish) make such tracking much more challenging. This is due mainly to the fact that the relevant contexts of multiple inhabitants in the same environment are often inherently correlated and thus inter-dependent on each other. Therefore, the learning and prediction (decision making) paradigm needs to consider the joint (simultaneous) location tracking of multiple inhabitants [10]. In the following, we present two different algorithms (cooperative and non-cooperative) for multiple inhabitant cases.

2.1 Multiple Inhabitant Location Tracking

As mentioned earlier, the multiple inhabitant case is more challenging. Mathematically, this can be observed from the fact that conditioning reduces the entropy [2]. In [10] we proved that optimal (i.e., attaining a lower bound on the joint entropy) location tracking of multiple inhabitants is an NP-hard problem.

Cooperative Reinforcement Learning Algorithm: Assuming a cooperative environment, a game theory based reinforcement learning policy was proposed in [9] for location-aware resource management in multi-inhabitant smart homes. This approach describes an algorithm for a rational and convergent cooperative action learner. The basic idea is to vary the learning rate used by the algorithm so as to accelerate the convergence, without sacrificing rationality. In this algorithm we have a simple intuition like “learn quickly while predicting the next state incorrectly”, and “learn slowly while predicting the next state correctly”. The method used here for determining the prediction accuracy is by comparing the current policy’s entropy with that of the expected entropy value earned by the cooperative action over time. This principle aids in convergence by giving more time for the other inhabitants to adapt to changes in their strategy that at first appear beneficial, while allowing them to adapt more quickly to the other inhabitants’ strategy changes when they are harmful.

Non-cooperative Nash H -Learning Algorithm: Hypothesizing that every inhabitant wants to satisfy his own preferences about activities, we assume he behaves like a selfish (non-cooperative) agent to fulfill his own goals. Under this circumstance, the objective of the system is to achieve a suitable balance among the preferences of all inhabitants residing in the smart home. This motivates us to look into the problem from the perspective of non-cooperative game theory where the inhabitants are the players and their activities are the strategies of the game. Moreover, there can be conflicts among the activity preferences. Our proposed game theoretic framework [10] aims at resolving these conflicts among inhabitants, while predicting their activities (and hence locations) with as much accuracy as possible.

The proposed Nash H -learning (NHL) algorithm [10] significantly enhanced the Nash Q -learning algorithm [6] in that it captures the location uncertainty in terms of entropy at each and every step of the inhabitants' mobility path. Thus, in our case, Nash H -value is determined to satisfy both Nash condition as well as our imposed entropy (uncertainty) minimization constraint. We assume that the inhabitants are fully rational in the sense that they can fully use their location histories to construct future routes. As a result, the decision making component should not directly repeat the actions of the inhabitants but rather learn to perform actions that optimize a given reward (or utility) function. For this optimization, our proposed entropy learning algorithm (NHL) learns a value function that maps the state-action pairs to future reward using the entropy measure, H . It combines new experience with old value functions to produce new and statistically improved value functions. The proposed multi-agent Nash H -learning algorithm updates with future Nash equilibrium payoffs and achieves a Nash equilibrium such that no inhabitant is given preference over others. This results in more accurate prediction of contexts and better adaptive control of automated devices, thus leading to a mobility-aware resource (say, energy) management scheme in multi-inhabitant smart homes. Experimental results demonstrate that the proposed framework adaptively controls a smart environment, significantly reduces energy consumption and enhances the comfort of the inhabitants.

3 Ambiguous Context Mediation

In this section we discuss a framework that fuses data from disparate sensors, represents context state (activity) and reasons efficiently about this state, to support context-aware health care services that deal with ambiguity. For such environments with ambiguous contexts, our goal is to build a framework that resolves information overlap conflicts, and also ensures the conformance to the application's quality of context (QoC) bound based on an optimal sensor configuration. For this purpose, we propose a Dynamic Bayesian Network (DBN) based model [11] in which the sensed data is used to interpret context state through the fusion process. The use of DBNs as our baseline sensor fusion mechanism reflects this analogy whereas an information theoretic reasoning selects the optimal context attributes (sensor data) value to minimize the state ambiguity or error. We build a system using various kinds of SunSPOT [15] sensor for sensing and mediating user context state or activity. Experimental results demonstrate that the proposed framework is capable of adaptively enhancing the effectiveness of the probabilistic sensor fusion scheme and patient's situation prediction by selectively choosing the sensor corresponds to the most economically efficient disambiguation action.

3.1 Context-Aware Data Fusion

A characteristic of a sensor-rich smart health care environment is that it senses and reacts to *contexts*, information sensed about the environment's inhabitants

and their daily activities, by providing context-aware services that facilitates the inhabitants in their everyday actions. Here we develop an approach for sensor data fusion in context-aware health care environment considering the underlying space-based context model and a set of intuitions it covers. In the case of context-aware services, it is really difficult to get an accurate and well defined context which we can classify as ‘unambiguous’ since the interpretation of sensed data as context is mostly imperfect and ambiguous. To alleviate this problem, we propose a DBN model based on which we design a context-aware data fusion framework to reduce this ambiguity as much as possible during the situation inference (e.g., patient’s behavior or sickness) process.

3.2 Information Theoretic Reasoning

We introduce a formalism for optimal selection of sensor parameters for state estimation. The optimality is defined in terms of reduction in ambiguity or error in the state estimation process. The main assumption is that the state estimation becomes more reliable and accurate if the ambiguity/error in the underlying process can be minimized. We investigate this from an information theoretic perspective [2] where information about the context attribute is made available to the fusion center by a set of smart sensors. The fusion center produces an estimate of the state of the situation based on the intelligent analysis of the received data. We assume that the noisy observation across sensors are independent and identically distributed (i.i.d) random variables conditioned on the binary situation \mathcal{R} (assumed binary \mathcal{R} for ease of modeling). Now each sensor attribute a_i has a source entropy rate $H(a_i)$. Any sensor wishing to report this attribute must send $H(a_i)$ bits per unit time which is the entropy of the source being measured assuming that the sensor is sending the ‘exact’ physical state. So, the problem is to minimize the error (or keep it within a specified bound), while not exceeding the shared link rate \mathcal{Q} . Thus by maximizing the posteriori detector probability we can minimize the estimation error of the random variables to accurately reconstruct the state of the situation.

Problem. *Let B be the vector of sensors and A be the set of attributes, then imagine a matrix $(B \times A)$ where $B_{mi} = 1$ when sensor m sends attribute a_i . Then, the goal is to find a matrix $(B \times A)$ within the capacity constraint \mathcal{Q} which minimizes the estimation error of the situation space.*

$$\sum_m \sum_i H(a_i) * B_{mi} < \mathcal{Q} \quad \text{and} \quad \text{minimize} [P_e = P\{\tilde{\mathcal{R}} \neq \mathcal{R}\}] \quad (1)$$

where $\tilde{\mathcal{R}}$ is an estimate of the original state \mathcal{R} .

We observe that the Chernoff information [2] at the fusion center is monotonically increasing in the number of sensors for a fixed decision rule π . State estimation error can be minimized by augmenting the number of sensors in π until the capacity constraint \mathcal{Q} is met with equality. The strategy π being arbitrary, we conclude that having \mathcal{Q} identical sensors, each sending one bit of information

is optimal in terms of reducing the state estimation error. This configuration also conveys that the gain offered through multiple sensor fusion exceeds the benefits of getting detailed information from each individual sensor.

4 QoINF-Aware Context Determination

Here we suggest a formal approach [12] to minimum-cost (cost defined in terms of such metrics such as energy or bandwidth), *continuous* determination of an individual's context in smart environments. Our framework presupposes the use of an event-driven data framework, where each individual sensor is associated with a *tolerance range*, indicating the amount of imprecision that can be tolerated by the monitoring application. Central to our model is the notion of a Quality of Inference (QoINF) specification, defined as the error probability in estimating a context state, given the imprecision in the values of the contributing sensors. There are two main observations driving this work: a) Smart environments typically contain several sensors, with a particular activity context capable of being estimated to varying degrees of accuracy via data from different sensors. More importantly, the accuracy of the inferred context increases with the use of a progressively larger sensor set (often with different modalities). As a simple example, a combination of data from a body-worn accelerometer and ceiling mounted motion sensors provides a more accurate estimation of whether 'a person is immobile after a fall', compared to deductions based solely on each individual sensor. b) The quality of the inferred context is not just a function of the chosen sensors, but also of the permitted inaccuracy in the sensor values. In general, the larger the uncertainty in the precise value of a data sample, the lower the inferencing accuracy. Broadly speaking, this part of the work advocates the development of a formal methodology for answering the following question: "Given an application-defined specification of a minimal acceptable QoINF value, how do we compute both the optimal set of sensor data streams that are needed for inferencing, and the optimal tolerance ranges permissible for each selected sensor?"

4.1 QoINF Cost Optimization

Our computing infrastructure consists of a declarative query processing engine that takes application bounds on $QoINF_{min}$ as input and optimally 'tasks' the individual sensors to provide the necessary inputs to a context estimator engine. The actual (optimal) parameters will depend on the specific cost (e.g., energy or reporting frequency) that we seek to minimize and the structure of the $QoINF(.)$ function. In this section, for specificity, we focus on minimizing a measure of the average *communication overhead*.

Average Reporting Cost Optimization: One natural cost to optimize is the communication overhead incurred by the sensor in reporting its values to the Context Estimator component. Let us denote the average update cost (communication overhead) of sensor s_i , given a tolerance range q_i as $c_i(q_i)$. Intuitively,

c_i is a decreasing function of q_i , since the communication cost would be higher (more frequent reports) for smaller tolerance ranges. In a setting where the sensor data traverses multiple hops to get to the Context Estimator Engine, the update cost is also proportional to h_i , the length of the uplink path from sensor s_i to the Aggregation Engine. If the underlying data samples evolve as a random-walk model, we have $c_i(q_i) \propto \frac{h_i}{q_i^2}$. In this case, the resulting cumulative cost function is given by:

$$COST(\theta, q_\theta) = \kappa * \sum_{i \in \theta} \frac{h_i}{q_i^2} \quad (2)$$

where κ is a scaling constant and h_i is the hop count.

If the set of sensors to be used, i.e., the set θ is given, then the problem of optimally computing the q_i s can be represented by the Lagrangian:

$$\text{minimize } \sum_{i \in \theta} \frac{h_i}{q_i^2} + \lambda \times [QoINF_C(q_1, q_2, \dots, q_\theta) - QoINF_{min}]. \quad (3)$$

Finding an exact solution to Expression 3, for any arbitrary $QoINF(\cdot)$ is an NP-complete problem [5]. For the general case of a function $QoINF(\cdot)$, the only solution to determine the most optimal set of sensors (i.e., $\hat{\theta}$) is to iterate over all the $2^S - 1$ elements of the power-set of S . While a completely arbitrary $QoINF(\cdot)$ function requires a brute-force search, there are certain forms of $QoINF(\cdot)$ that prove to be more tractable and lend themselves to more efficient optimization heuristics [14].

The above presented an initial design of a formal framework for energy-efficient determination of activity or physiological contexts in assisted living environments. The key idea was to express the accuracy of context estimation through a $QoINF(\cdot)$ function that captures the dependence of estimation accuracy on both the set of selected sensors and their specified tolerance ranges. Besides implementing and empirically quantifying the validity of the proposed framework, our future work will investigate different forms of $QoINF(\cdot)$ functions that might lend themselves to provably linear-time strategies for computing the optimal $(\theta, \{q_\theta\})$ combination. Similarly, for our initial choice of the quality of inference function, the proposed sensor selection heuristic requires evaluation. Properly addressing these issues will lead to a smart environment with reliable functionality that improves the quality of life for inhabitants and our communities.

5 Future Work and Open Problems

There are many ongoing challenges that researchers in this area continue to face. The first is the ability to handle multiple inhabitants in a single environment. While this problem is addressed from a limited perspective [10], modeling not only multiple independent inhabitants but also accounting for inhabitant interactions and conflicting goals is a very difficult task, and one that must be

further addressed in order to make smart environment technologies viable for the general population. Similarly, we would like to see the notion of “environment” extend from a single setting to encompass all of an inhabitant’s spheres of influence. Many projects target a single environment such as smart home, smart office, smart mall, smart airport. However, by merging evidence and features from multiple settings, these environments should be able to work together in order to customize all of an individual’s interactions with the outside world. As an example, how can we generalize intelligent automation and decision-making capabilities to encompass heterogeneous smart spaces such as smart homes, vehicles, roads, offices, airports, shopping malls, or hospitals, through which an inhabitant may pass in daily life. An interesting direction that researchers in the future may consider is not only the ability to adjust an environment to fit an individual’s preferences, but to use the environment as a mechanism for influencing change in the individual’s behavior or life style. For example, environmental influences can affect an individual’s activity patterns, his mood, and ultimately his state of health and mind. Finally, a useful goal for the research community is to define figure of merits, performance metric, and benchmarks to evaluate and compare different smart environments. While performance measures can be defined for each technology, those for an entire smart environment still need to be established. This can form the basis of comparative assessments and identify areas that need further investigation.

References

1. Bhattacharya, A., Das, S.K.: LeZi-update: An information theoretic approach for personal mobility tracking in PCS networks. *ACM Wireless Networks (WINET)* 8(2), 121–137 (2002) (This is an extended version of the Best Paper in ACM MobiCom 1999)
2. Cover, T.M., Thomas, J.A.: *Elements of Information Theory*. John Wiley, Chichester (1991)
3. Cook, D.J., Das, S.K.: *Smart Environments: Technology, Protocols and Applications*. John Wiley & Sons, Chichester (2004)
4. Das, S.K., Cook, D.J., Bhattacharya, A., Heierman, E., Lin, T.Y.: The Role of Prediction Algorithms in the MAVHome Smart Home Architecture. *IEEE Wireless Communications (Special Issue on Smart Homes)* 9(6), 77–84 (2002)
5. Deshpande, A., Guestrin, C., Madden, S., Hellerstein, J.M., Hong, W.: Model-based Approximate Querying in Sensor Networks. *Int’l Journal on Very Large Data Bases (VLDB Journal)* 14(4), 417–443 (2005)
6. Hu, J., Wellman, M.P.: Nash Q-Learning for General-Sum Stochastic Games. *Journal of Machine Learning* 4, 1039–1069 (2003)
7. Roy, A., Das, S.K., Basu, K.: A Predictive Framework for Location Aware Resource Management in Smart Homes. *IEEE Transactions on Mobile Computing* 6(11), 1270–1283 (2003)
8. Roy, N., Roy, A., Das, S.K., Basu, K.: A Reinforcement Learning Framework for Location-Aware Resource Management in Multi-Inhabitant Smart Homes. In: *Proc. of 3rd International Conference on Smart Homes and Health Telematic (ICOST)*, pp. 180–187 (July 2005)

9. Roy, N., Roy, A., Basu, K., Das, S.K.: A Cooperative Learning Framework for Mobility-Aware Resource Management in Multi-Inhabitant Smart Homes. In: Proc. of IEEE International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous), pp. 393–403 (July 2005)
10. Roy, N., Roy, A., Das, S.K.: Context-Aware Resource Management in Multi-Inhabitant Smart Homes: A Nash H-learning based Approach. In: Proc. of IEEE Int'l Conf. on Pervasive Computing and Communications (PerCom), pp. 148–158 (March 2006) (Mark Weiser Best Paper Award); (Extended version appeared in Pervasive and Mobile Computing Journal. 2(4), 372–404 (November 2006)
11. Roy, N., Pallapa, G., Das, S.K.: A Middleware Framework for Ambiguous Context Mediation in Smart Healthcare Application. In: Proc. of IEEE Int'l Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob) (October 2007)
12. Roy, N., Misra, A., Das, S.K.: Efficient Long-Term Quality-of-Inference (QoINF)-Aware Context Determination in Pervasive Care Environments. In: Proc. of ACM SIGMOBILE Workshop on Systems and Networking Support for Healthcare and Assisted Living Environments (HealthNet 2008) held in conjunction with MobiSys 2008 (June 2008)
13. Roy, N., Pallapa, G., Das, S.K.: An Ontology-Driven Ambiguous Contexts Mediation Framework for Smart Healthcare Applications. In: Proc. of Int'l Conf. on Pervasive Technologies Related to Assistive Environments, PETRA 2008 (July 2008)
14. Roy, N.: A Context-aware Learning, Prediction and Mediation Framework for Resource Management in Smart Pervasive Environments, Ph.D Thesis (August 2008)
15. SunSpotWorld-Home of Project Sun SPOT, www.sunspotworld.com

Implicit Middleware

A Ubiquitous Abstract Machine

T. Riedel¹, M. Beigl², M. Berchtold¹, C. Decker¹, and A. Puder³

¹ TecO, University of Karlsruhe

² Distributed and Ubiquitous Systems, University of Braunschweig

³ San Francisco State University

Abstract. This paper introduces an approach for abstracting access to functionality in Pervasive Computing systems where very different types of devices co-exist. Tiny, resource-poor 8-bit based wireless embedded sensor nodes use highly fragmented programming, with code distributed over possibly hundreds of nodes. More powerful devices as mobile, handled devices, laptops or even server use coarse-grained distribution. The Implicit Middleware approach provides a way to both unify and simplify middleware for Pervasive Computing systems, by means of transparently distributing functionality in the system and making them context aware. The approach ensures optimized run-time behavior and adaptation to the system landscape. We also present an implementation using the XMLVM representation for code generation, and an evaluation running on PCs, J2ME CLDC 1.0 compatible 32Bit sensor nodes and 8Bit-MCU based nodes with an optimized light-weight VM.

1 Introduction

As predicted by Marc Weiser, computing systems have changed over the last decade in terms of quantity and their degree of specialization to a task. This has severe consequences on the way distributed systems are build: While on distributed server systems the distribution of resources (computing load, memory etc.) was dominant, with the advent of small mobile computing systems we separate between user driven concerns - user interface or user specific application logic - and server driven concerns. Smart items, physical things coupled with embedded computational intelligence and a communication interface, allow pushing logic further towards the edge of the real world. Both web browsers and networked sensors have in common that efficiently programming and seamless integration of such dedicated platforms into software infrastructures is challenging. But, distributed execution and "logic on the item" can reduce response times of applications and improve the overall efficiency by employing all available resources in the network substantially [1].

Such change also affects the way functionality is distributed among the participating computing systems. Today, the programmer is in charge of identifying and defining at least the parts of the software that should be distributed. This is done e.g. by defining the interfaces of software parts in an IDL. Additionally the

programmer or designer often has to define how to distribute the functionality - by putting the software onto servers and clients - and where the software parts should be run. Knowing how the target platforms look like this design decision can easily be made. With traditional computing this was the case: Distributed systems are separated with having in mind if the software runs on a supercomputer, a database server, a PC or a small mobile device. Thus, dependent on the type of systems different software architectures and distribution interfaces were chosen. With the advent of small, very diverse type of ubiquitous computing systems like sensor nodes, such an approach will be not suitable anymore. We assume that the design space of these devices will be huge and will stay huge. The exponential grow of semiconductor capabilities over time leads to a grow in resources and thus to a kind of unification of device capabilities in traditional computer systems. Functionality of ubiquitous systems and sensor nodes is additionally restricted by the capabilities of their interface to the real world - the sensors - and their energy supply - battery or parasitic. Both resources are known to not grow exponentially, thus a unification of such devices cannot be expected.

Using the same distribution strategy for ubiquitous as for traditional distributed systems would require the design of specialized distributed systems for all types of hardware. But the design space, regarding parameters like energy consumption, response time and hardware constraints, is huge. We propose the use of *implicit* middleware (*IM*) as a solution for this problem. In contrast to explicit middleware as described above, *IM* should not require a programmer to know about the underlying functionality of the system, or even to describe interfaces to be used by the middleware system at all. In some sense, *IM* works like a distributed virtual machine, by distributing functionality among available components automatically. We think, that this makes *IM* a concept especially suitable to a Pervasive Computing environment. *IM* abstracts completely from the distribution and allows software to run efficiently on any hardware landscape, without the need for redesign, and without pre-knowledge by designer and programmer. Programs are designed as monolithic application for an abstract machine. In contrast to classical approach the middleware becomes part of the program semantics thus is *implicit*. To enable distribution of functionality, we separate the application into parts, and use code transformation on the parts before transferring them to the target system automatically. This considerably increases the level of abstraction and leaves room for optimization. By “delaying” the partitioning of the software until the final deployment, we can calculate on optimal distribution based on the parameters of the system landscape.

To show the feasibility of our system, we present an implementation of an *IM*, which is based on stub code generation, transformation and optimization on compiled binary Java applications using the XMLVM [2], the Eclipse Test and Profiling Toolkit [3] and the Zuse Institute Mathematical Programming Language [4]. Although distribution frameworks often have the problem of high runtime overhead, we can show that our approach niftily works around those problems by using static code generation techniques at deploy time. A resource optimized implementation for the extremely low-power Particle Computer [5] wireless sensor nodes shows the feasibility for using the approach to run a sensing

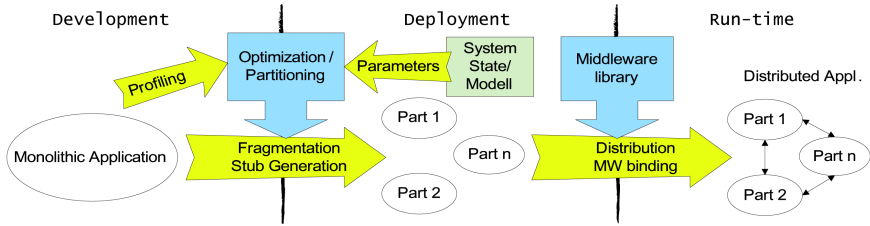


Fig. 1. Design of the Implicit Middleware

application with graphical user interface distributed between a PC and resource efficient networked sensors.

2 Design of an Implicit Middleware

The goal of the *IM* concept is to allow application development independent of target platforms and distribution semantics. The *IM* therefore takes a monolithic application and transforms, partitions and distributes it optimally regarding parameters of the underlying platforms and networks. At the current status, the concrete application of the *IM()* concept is restricted to support deployment time processes only. In general, the concept is extensible to run-time, which is looked at in future work.

Figure 1 shows the general architecture of the *IM*. We start with the pre-compiled monolithic application that we want to deploy. Two parameters influence the optimal distribution: The application profiling information provides additional design decision information, while the system state and model contains information about the current contextual state of all system parts that should be involved in the application. Both information sources are used to partition the application. At the cuts we replace the bindings to other parts by generated stub code that instead binds the application against the middleware layer. Following we use specific platform transformations to generate target code for both the application parts and middleware stubs. The platform code is then deployed to different hosts according to the distribution. At runtime the middleware library ensures that all parts are networked together executing the application collaboratively.

The middleware layer is the runtime component that is needed to be present for every platform that should support the *IM*. As generally the *IM* follows a generative approach this part contains the minimal generic functions that have to be provided by all middleware services on any of the participating devices in a as a platform specific component. Because this part is deliberately kept as lean as possible, it is easily portable with only minimal effort. The middleware layer implements the remote invocation of the following three method types via synchronous RPC: instantiation, virtual methods and static methods. At runtime the *IM* provides a marshalling API similar to Sun XDR that is optimized to be used by the code generation. Because we generate stub and dispatcher at the same time we know the message layout and can rely on the typing of the message.

Only the class type, method identifier must be encoded via locally unique ids into the message. The call parameters are de/encoded via generated serialization calls. As we operate on byte code level the arguments are subsequently pushed to/popped from the VM stack so that we immediately obtain the correct stack layout for dispatching the local call.

3 Code Transformation

The automatic generation of stub code is one of the key concepts of the *IM*. Stubs realize the Interface between application code and the middleware library. We achieve a high degree of transparency, because the developer does not have to implement distribution aspects manually. The stub proxies the original class and interacts with the middleware. To achieve maximum performance and high portability, the overall system is split into three blocks. The Specific part contains all platform specific parts such as bindings to communication, the Generic part contains a minimalistic middleware layer (see above), while the Generated parts contains the complete interaction with the middleware. Not building our distribution logic on runtime changes allows us to realize a minimalistic and lightweight middleware concept. This way the whole system does not rely on any external features like reflection beyond the byte-code instructions. The *IM* thus can either run on any node that supports only a minimal virtual machine instruction set.

3.1 Platform Independent Stubs

If we want to enable remote access via stubs, a few steps have to be taken. First, all classes that access object or class variables directly have to be rewritten using getter and setter methods. If the code is cleanly written object-oriented code this step is often not necessary. Then we generate a stub class as proxy for the original class byte code. In order to generate stub code from compiled Java classes we:

1. remove all object/class variables from the code.
2. add an object id field that can be used to link against the remote object
3. add a constructor to instantiate a stub with a remote object id
4. strip all code from the methods keeping the declarations and
5. fill the methods with generated code that creates a remote call object, adds all arguments and starts a call

By replacing method code with stub-methods we ensure that the stub classes retain all properties of the original classes. This way all links from the outside to this class stay intact.

To perform those Transformations we transform the binary Java code to an XMLVM representation. The advantage of this step is that transformations can operate on the Document Object Model (DOM) of XML. Therefore efficient transformation chains using XSLT or JDOM can be used on the tree structure. Furthermore XMLVM offers code generator for different target languages like JavaScript or C++. This way the generated stub code as well as the class files can easily be ported to non-Java enabled platforms.

3.2 Platform Independent Dispatching

In order to invoke methods remotely we need a dispatching mechanisms that fits our stub code on the remote side. The dispatching mechanism can also be designed completely using the Java byte code representation XMLVM. By not using reflection features at runtime, the code firstly stays efficient and secondly is highly portable just as the stub code.

Because Java byte code and thus the basic instruction set of the XMLVM does not support dynamic invocation via arguments, the most simple and generic way of generating a dispatcher is using a jump table for all classes. Because stub code and dispatcher code are generated at the same time, we have no problems assigning perfectly hashed keys as local identifiers.

3.3 Platform Dependent Code

Although the XMLVM byte code we get after transforming our application could be interpreted natively; this would generate a considerable overhead. Therefore, the XMLVM code is transferred into different target languages. Those targets do not have to be known beforehand, but can be configured just in time for code deployment and distribution. As already described in [6] an elegant way to generate target code is applying different XSLT transformations to the XMLVM representation. By default we transform the code back to a byte code representation. This method can however also be used to generate JavaScript code to execute in web browsers or C++ code for compilation to native machine code.

4 Profiling and Optimization Framework

Up to this point we did not talk about implementation of one of the most important aspects of the *IM*: optimized distribution. While the techniques described before enable the arbitrary distribution of a monolithic application, the ultimate goal is using this advantage to increase application performance. As the application performance heavily depends on the trade-off between computation speed and execution time, the distribution of an application over multiple platforms and networks affects overall execution speed, responsiveness and energy consumption. In order to calculate an optimized distribution of an application we have to estimate the run-time cost for a specific instantiation.

Given is a set of classes in an application A $C_A = \{c_1, \dots, c_n\}$ of platforms $P_S = \{p_1, \dots, p_n\}$ that are available for deployment in the computing system S . The *IM* now tries to solve the optimal mapping of A on S , by finding an optimal of C_A on P_S that is defined by the mapping relation M . The characteristic function $c_M(c, p)$ of M evaluates to 1 if class c is mapped on platform p and otherwise. For our subsequent integer linear optimizations we use a binary variable to partially define $c_M(c, p) = \mu_{c_i, p_j}; c = c_i, p = p_j$ over $C_A \times P_S$. To obtain an optimal mapping we continue with a minimization of a cost function that depends on the different assignments of μ_{c_i, p_j} .

We base this cost function on the different execution and communication cost of the various mapping. To do this we first need to define a function $t_e(A, c, p)$

that results in the total execution time of code inside a class on a given platform c for a given application A . Because this obviously is not computable for most applications A , we base this estimation on a program trace τ_A using an estimator $\hat{t}_e(\tau_A, c, p)$. Accordingly we define a function to estimate the (additional) communication on RPCs $\hat{t}_c(\tau_A, c, p, c', p')$ between a class c on platform p and a class c' on platform p' .

In our implementation we use output from the Eclipse Test and Performance Tools Platform (TPTP) [3] to generate and initialize constraints and the cost function. The TPTP smoothly integrates into existing software development cycles via the Eclipse IDE and uses a portable XML output format for interfacing. From the trace we generate a linear program that we can solve automatically. The functions $\hat{t}_{e,A}$, which we obtain from currying $\hat{t}_c(\tau_A)$, and respectively $\hat{t}_{i,c,A}$ are precomputed. We use the the Zuse Institute Programming Mathematical Language [4] with the soPlex solver, to implement the optimization. Currently we assume a linear speed scaling between the platforms as well averaged communication cost per call and platform combination. Those assumption can hold when not using (just in time) compilation techniques and staying below the maximum bandwidth of the network. Although more complex optimization strategies and cost models could be integrated into the framework, the current granularity is sufficient to achieve major improvements while providing simple model configuration.

In our current *IM* implementation we use the following composite objective to optimize the execution speed of an application A :

$$\begin{aligned} & \text{minimize cost} : \sum_{c,p \in C_A \times P_S} : \hat{t}_{e,A}(c,p)\mu_{c,p} + \\ & \sum_{c,p,c',p' \in C_A \times P_S \times C_A \times P_S} : \hat{t}_{c,A}(c,p,c',p')\mu_{c,p}\mu_{c',p'} \end{aligned}$$

In order to linearize this objective by replacing the term $\mu_{c,p}\mu_{c',p'}$, we define a second set of binary variables $\mu'_{c,p,c',p'}$, which we interlink via a linearized logical implication:

$$\forall c,p,c',p' \in C_A \times P_S : \mu'_{c,p,c',p'} \geq \mu_{c,p} + \mu_{c',p'} - 1 \wedge \mu'_{c,p,c',p'} \leq \mu_{c,p} \wedge \mu'_{c,p,c',p'} \leq \mu_{c',p'}$$

Additionally there are other constraints that need to be considered. These include the availability graphical interfaces or enough data or program memory. As a simple example we may fix a certain hardware driver class to a platform by a constraint *subto fix* : $\mu_{c,p} = 1$. Last but not least we ensure that the mapping is complete, i.e. each class is mapped once, by demanding *subto mapall* : $\forall c \in C_A : 1 = \sum_{p \in P_S} \mu_{c,p}$.

These constraints open up the optimization space that includes all possible distributions of classes over the platforms. Minimizing according to the objective under these constraints gives us a classical constrained linear optimization problem that can be solved by the *IM*. The output of the optimization is a mapping relation M that meets the constraints with the least costs according to our estimation function. It represents the optimal code distribution that can be achieved on class granularity regarding a given parametric model of the target platforms. If an application is deployed via the *IM* uses information about the current system landscape and a given program trace to generate M and partition, as well as distribute, the application accordingly.

5 Evaluation

The 2/32 Particle Computer sensor node [5] includes a Microchip PIC18F6720 micro controller. This low power MCU has an instruction cycle of $0.2\mu s$ and includes only 4 Kbytes of RAM and 128 Kbytes of code memory. In contrast to directly executed machine code the 512K of external Flash memory can be additionally used as program memory holding Java classes. The current implementation of the byte code interpreter occupies 60KB of code memory and 1.5 Kbytes of memory. Additionally 45 Kbytes of code memory and 0.5kbyte of RAM are dedicated to the low-level native API for the sensor node with basic operating system features and the proprietary RF functionality including message buffers. This leaves 1.5 Kbytes of heap memory that can be used by any user program running on top of the ParticleVM. In order to prove high portability we additionally ported the *IM* to the SunSPOTS platform, which comprises a 32bit 180 MHz Arm9, 512 Kbyte RAM, 4 Mbyte Flash, built-in sensors and a IEEE 802.15.4 compatible radio and features a fully J2ME CLDC 1.0 compatible VM.

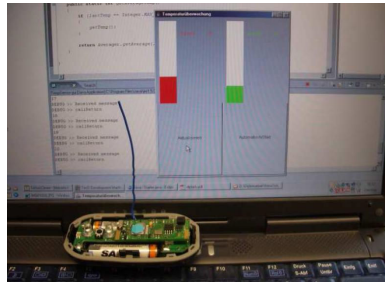


Fig. 2. Distributed Sensor and GUI program

The overhead generated by the distribution through the *IM* generated code overhead on the Particles platform relative to the number of classes n_c , methods n_m , method arguments n_a is $14 + 10(n_c + n_m) + 8n_a$ bytes of instructions for the dispatcher using conditional branching instruction and $48_c + 8n_m + 4n_a$ for all stubs. Additionally $4n_o + 8n_{o'}$ bytes of RAM are needed additionally for instantiated objects n_o and remotely referenced objects $n_{o'}$. The numbers were calculated on the basis of the generated code for the Particle sensor nodes using only integer or reference type arguments. Marshalling and RPC libraries are independent of the distributed program and consume only 72 and 114 byte of program memory respectively. The RPC handling additionally has an 18-byte RAM footprint.

On the SunSPOT platform the overhead is considerably larger, because of the less optimal byte-code encoding. Here the constant part of the middleware occupies 37 Kbyte. The dynamically generated stub class uses as much as 1.3 kbyte of ROM for a class with 4 Methods. The dispatcher for 44 Methods in 6 different classes has a size of 4,13 kbyte. In contrast to the Particle platform,

which only supports a single thread, the runtime overhead on the SunSPOT currently is dominated by the thread creation overhead. We measured an average roundtrip time for a simple RPC of 513 ms, while only 42ms could be contributed to the communication overhead and the program execution.

5.1 Optimization

In order to evaluate the optimization framework we configured the optimizer using different cost functions on a small example program. The program depicted includes:

- a temperature sensor that resides on a sensor node
- a vibration sensor also bound to a sensor node platform
- a processor intensive Fourier analysis of the vibration data to isolate frequencies
- an alarm system that decides based on different contexts what alarm level to raise

Further we assume three possible target platforms to be available. Based on this data we describe different optimization goals. First we optimize the application regarding latency, resulting from remote calls. To simplify the view on the problem we assume equal latencies for calls between all platforms. The *IM* optimized the system in a way that the sensor post-processing was done on the same platform as the sampling reducing communication resulting in Distribution V1. This distribution represents what you expect to be a sound choice if your sensor network platform is able to calculate the Fourier transformation fast enough. That might be true for a 32-bit 180 MHz platform like the SunSPOT, but is different for a platform like Particle Computers that use only 8-bit MCU, which executes machine instructions at 5MHz. To illustrate the potential of the *IM* we took the same application and changed the optimization parameters accordingly. If we do that we can see that the sensor nodes here only execute the sensing as well as the computation inexpensive task control task for the temperature sensing. The computation-intensive FrequencyController was shifted on a 2.6 GHz system that was also available. We expect this to be the fastest possible distribution variant, although more communication is needed now.

To illustrate the performance gains achieved by employing the optimization possibilities of the *IM* we compare the cost functions for the latency-biased optimization in table 1a and for an optimization based on the different execution speeds in table 1b. You can see that we assume no cost for local calls. This is compliant to our model as the *IM* leaves code for local invocation untouched, so that there is no middleware overhead. Although both distributions could make sense at design time we can see that for latency the *IM* was able to achieve a 29% speed-up. In contrast to that the same distribution would have lead to an 87% slowdown if instantiated on Particle Computer sensor nodes. Those results indicate the potentials of the proposed scheme. The optimization problem for this simple example already expanded to 342 variables and 980 constraints, which however can be solved using soPlex in below 200ms on an 1.66GHz Intel CPU.

Table 1. Distribution Cost Comparison

call	cost _{V1}	cost _{V2}
TempTask - TempSensor	0.0	0.0
TempTask - Alarm	13.0	13.0
FreqContr - VibrSensor	0.0	7.0
FreqContr - Alarm	2.0	0.0
FreqContr - Fft	0.0	0.0
Starter - TempTask	2.0	2.0
Starter - FreqContr	0.0	0.0
overall costs	17.0	22.0

(a) Latency

call	cost _{V1}	cost _{V2}
TempTask	27.79	27.79
TempSensor	0.03	0.03
VibrSensor	3.71	3.71
FreqContr	24.08	0.19
Starter	0.27	0.01
Alarm	0.01	0.01
Fft	2.80	0.02
overall costs	58.68	31.74

(b) Execution

6 Related Work

Quite a number of other middleware systems are available in general and for ubiquitous systems particularly. For sensor network application active distributed databases like TinyDB [7] or Cougar [8] commonly allow the distributed execution of queries and functions. In contrast to our work those approaches see assume the same functionality on all devices together with a more powerful central data sync, which acts as a controller. Although quite different in their functionality also tuple space architectures like used in Jini [9] provide additional middleware abstraction to the programmer. In contrast to other approaches the *IM* aims at total transparency by unifying execution and distribution abstractions.

A unified abstraction is also provided by distributed programming languages like the functional language Regiment [10] or the distributed Prolog-like Cooperative Artifact System [11]. Although those are not middleware systems in a strict sense, they provide distribution features intrinsically. In contrast to our work they use very domain specific languages to solve the problem, which make it difficult to port existing applications.

The system in [12] is similar to our approach in enabling the partitioning of application for networked sensor platforms. While being limited to TinyOS systems, it closely integrates with the compiler and therefore does not have the flexibility of our approach, which is decoupled from the development tool chain via the XMLVM and targets wider range of platforms. For traditional computer systems the distribution framework j-orchestra [13] offers automatic partitioning on pure java systems on the basis RMI. The use of Java reflection mechanisms inside the run-time system makes it not fitted for embedded systems such as CLDC 1.0 platforms. In contrast to j-orchestra our work is build around the principles of minimalism to support a wide range of ubiquitous computing platforms. The generated marshalling code is the basis for a much more portable and lightweight underlying middleware layer. Most importantly our work includes an optimization phase to adapt the application deployment dynamically. While both [12] and [13] focus on the technical aspects of transparent partitioning, our approach also implicitly generates the platform mapping based on a parametric system model, which makes it unique in this context.

7 Conclusion and Future Work

In this paper we presented a transparent middleware approach that uses code transformation techniques to distribute a monolithic application on multiple hybrid platforms. Especially the fact that it generates lightweight code and uses a minimal platform specific middleware layer makes it ideal for small networked embedded systems. Late optimization of the distribution before deployment allows us to adapt the software to perform optimal regarding our distribution model and the parameters of the underlying networks and hardware. Currently these models are limited to linear expectencies for computation and communication costs of a class-based partitioning. In the future more work will to be done in order to integrate other aspects of distributed computing into the *IM*. Generally we also plan to support real object migration to support dynamic reconfiguration at runtime and service mobility. As we recently have completed our JME CLDC 1.0 port of the *IM*, we plan to evaluate the system, using more realistic applications employing multiple alternative platforms. Settings with different sensor node platforms such as the Sun SPOTS and mobile phone platforms additionally to PC systems and Particle Computers will allow an interesting range of non trivial distribution scenarios to further evaluate the optimization phase.

By implementing and evaluating the *IM* for a low-power resource constraint platform like the Particle Computers sensor nodes and a CLDC 1.0 platform we have already shown in this paper, how we can integrate the small devices into high level application development. First user experiences already suggest that system support like the *IM* might radically change the attitude towards programming network sensor systems. Using the proposed techniques, we hope it will be possible to incorporate the skills of many developers that are used to high-level program design to develop for highly embedded ubiquitous systems.

References

1. Kubach, U., Decker, C., Douglas, K.: Collaborative control and coordination of hazardous chemicals. ACM Press, Baltimore (2004)
2. Puder: An xml-based cross-language framework. On the Move to Meaningful Internet Systems 2005: OTM Workshops (2005)
3. Eclipse: Eclipse test & performance tools platform project
4. Koch, T.: Rapid mathematical programming or how to solve sudoku puzzles in a few seconds. In: Operations Research Proceedings 2005 (2006)
5. Decker, C., Krohn, A., Beigl, M., Zimmer, T.: The particle computer system. In: 4th international symposium on Information processing in sensor networks (2005)
6. Puder: A code migration framework for ajax applications. Distributed Applications and Interoperable Systems (2006)
7. Madden, S.R., Franklin, M.J., Hellerstein, J.M., Hong, W.: Tinydb: an acquisitional query processing system for sensor networks. ACM Transactions on Database Systems (TODS) 30, 122–173 (2005)
8. Yao, Y., Gehrke, J.: The cougar approach to in-network query processing in sensor networks. ACM SIGMOD Record 31, 9–18 (2002)
9. Waldo, J.: Jini architecture for network-centric computing. Communications of the ACM 42, 76–82 (1999)

10. Newton, R., Welsh, M.: Region streams: functional macroprogramming for sensor networks. ACM Press, Toronto (2004)
11. Strohbach, G., Kortuem, K.: Cooperative artefacts: Assessing real world situations with embedded technology (2004)
12. Iwasaki, Y., Kawaguchi, N.: An Automatic Software Decentralization Framework for Distributed Device Collaboration. IEEE Computer Society, Los Alamitos (2007)
13. Tilevich, E., Smaragdakis, T.: J-Orchestra: Automatic Java Application Partitioning. Georgia Institute of Technology (2002)

Uncertainty Management in a Location-Aware Museum Guide

Pedro Damián-Reyes^{1,2}, Jesús Favela¹, and Juan Contreras-Castillo²

¹ Centro de Investigación Científica y de Educación Superior de Ensenada, Km. 107 Carretera Tijuana-Ensenada, CP 22860, Ensenada, B. C., México
{preyes, favela}@cicese.mx, juancont@ucol.mx

² Universidad de Colima, Facultad de Telemática, Av. Universidad #333, CP 28040, Colima, Col. México

Abstract. Uncertainty management remains a key issue in the design of reliable context-aware applications. Uncertainty is originated by both, the complexity associated to the acquisition of primary contextual information and imprecision in the derivation of secondary context. If uncertainty is not adequately identified and managed, it could render context-aware applications useless to the user. In order to assess the impact and utility of uncertainty management in a location-aware application, we conducted an experimental evaluation of a location-aware electronic guide for a museum. The experiment had two conditions: the first one directly uses location information estimated from the strength of the RF signal received in the mobile device from nearby WiFi access points, while the second implements a context uncertainty management mechanism being proposed. The impact was measured by comparing the number of location estimation errors and the number of user interactions required to operate the system. A questionnaire was used to measure the users' perception on ease of use, utility and trust on the application. The system was evaluated by 118 randomly selected visitors to the museum. Results suggest that uncertainty management helps improve the accuracy of the context estimate and the application performance. We also found that participants considered the implementation of context-aware technologies useful to assist them during their visits to the museum, regardless of the version of the application they used. However, ease of use and user trust is affected by the presence of uncertainty in the contextual information used by the application.

1 Introduction

Museum visitors are highly mobile and require specific information during their tour, which depends on their locations, interests and profile. These characteristics make museums good candidates for deployment of context-aware applications. At the same time, the use of images, video and audio descriptions trigger the sensitive memory and can enrich the experience of visitors [1].

However, context-aware application development is a complex task that involves several challenges, one of them being dealing with the uncertainty of contextual information [2, 3]. The uncertainty in context-aware applications can be generated by

the presence of uncertain, ambiguous or incorrect contextual information. Uncertainty directly affects user confidence and can cause the application to become useless to the user. Some methods have been proposed to deal with uncertainty in contextual information. Mainly Bayesian networks and ontologies have been used to tackle this problem [4-6], other approaches are based on the creation and chaining of rules [7, 8], and the use of fuzzy logic [5, 9]. However, a great amount of work by application developers is required and the aid of experts in the area is necessary. Moreover an assessment has not been done on the utility of using uncertainty management mechanism in a context-aware application.

In this paper we describe the main sources of uncertainty in contextual information, we proposed a procedure for creating an uncertainty management heuristic mechanism, and described an experiment to measured the impact and utility of the approach proposed for uncertainty management in a context-aware museum guide.

2 Uncertainty Management in Context-Aware Applications

Uncertainty arises when there is not clear knowledge of something, when there is the fear of error or there is doubt about what it is stated [10]. Penrod [11] defines uncertainty as a feeling that is related to the confidence that the individual has in himself and who has control over actions carried out. He states that the greater the confidence and control, the lesser the uncertainty. Li & Du [12] defined uncertainty as randomness and lack of clarity or accuracy of the concepts used in natural language. Several authors define imperfect information as a synonym of uncertainty, which includes vagueness in the sense that something is not well defined, imprecision as a failure in the specificity, incompleteness concerning the lack of pieces of information, the ambiguity that exists when it is not possible to distinguish between two alternatives, and the up-to-dateness of information concerning the possibility of exchange of information in the course of time. Error is another definition of uncertainty [6, 13], as well as inaccuracies in the devices and failures of the technology used [14, 15]. Inconsistency is another concept that has been associated with the uncertainty by several authors [5, 7, 16], with the meaning of the existence of information that contradicts or presents situations that violate established operating rules.

2.1 Uncertainty in Context-Aware Computing

In general, uncertainty in context may refer to three different notions: (a) uncertain context, (b) ambiguous context and (c) wrong context.

(a) Uncertain context. It occurs when the application malfunctions generating doubt in the user about the validity of the information or quality of service received. In general uncertain context affects the reliability of the system. This notion of uncertainty is generally caused by ambiguity or error in contextual information. For example, Benford et al. [15] describe situations where users expressed doubt about the information provided by the system, because certain areas showed jumps in the position of participants, disappearing and reappearing in different places; Beeharee & Steed [17] mentions that location-aware applications can generate uncertainty in the

user because the information submitted is sometimes difficult to interpret or does not correspond to the actual location of the user.

(b) *Ambiguous context*. It appears in different situations, such as in the definition of context using natural language. For example Beeharee & Steed [17] mention that the representation of the information is sometimes very abstract and difficult to relate to the real world, which hampers users' interpretation; Poole & Smyth [18] describe situations in which the non-specificity expressed in the concepts that define the objects can generate uncertainty. Ambiguous context is also presented when it comes from different sources. The use of heterogeneous technologies increases complexity in the acquisition and processing of context, while causing contradictions between the information provided.

The contradictions, violations of rules or inconsistencies are another situation in which it generates ambiguous context. For example, when information indicates that the user carries out different activities in the same time, when two or more actions carries out contrary operations on the same property of context in the same period of time, when it generates context that active different rules at the same time and each of them runs a different action or when there are contradictions generated by the occurrence of disjoint events [5], where different sources of generation create conflicting information [7].

(c) *Wrong context*. It refers to the accuracy and precision of the instruments, devices or technologies used to obtain context. The accuracy refers to the difference between the value obtained and the real value, while precision denotes the percentage of accurate measurement out of total number of measurements taken. In contextual information this refers to the accuracy with which the estimated context represents the reality of the physical world. For example, the Global Positioning System (GPS), one of the location technologies widely used by location-aware systems, has a large number of factors that affect the accuracy in obtaining a location [19], Benford et al. [15] mentioned that the lack or absence of information is another source of uncertainty, they indicate that presented situations where there is no complete information on the location the behavior application was erratic.

2.2 Uncertainty Classification

In order to simplify the identification of uncertainty, we have defined two types associated to context-aware applications:

Original uncertainty. It refers to the uncertainty that is inherent to the contextual element. It is closely linked with context generation. It appears when the application obtains the context from the sensors. The word 'sensor' refers not only to sensing hardware but also to every data source that provides usable context information. Sensors can be classified in three groups: physical, virtual and logical [20].

Derived uncertainty. It refers to the uncertainty generated as a result of the manipulation of contextual information and it is closely related to the applications use of context when it adapts its behavior to the new context.

This classification focuses only in two of the three previous mentioned notions of uncertainty: the ambiguous and the wrong context. The classification separates the uncertainty that is associated to the mechanisms for acquiring context from those used

to process it. The objectives of this separation are: (a) adaptability of the concept to most existing architectures for building context-aware applications, which generally establish a similar division in its architectures, (b) facilitate the work of uncertainty identification and treatment, and (c) implementation flexibility.

2.3 Managing Uncertain Information

In this section we describe a general procedure for the creation of an uncertainty management heuristic mechanism (UMHM) as an integral part of the systems development life cycle, including activities at the stages of analysis design and implementation of the application. The resulting mechanism is simple to implement and low-cost for its computational execution, and it is based on the establishment of rules that involve operation restrictions of the application. The proposed procedure consists of three main stages: identification, measurement and treatment of uncertainties.

Uncertainty identification. This involves establishing whether the application can produce original and/or derivative uncertainty, which requires a detailed analysis of the context-aware process. To find potential sources of uncertainty we study the way in which context is acquired and processed, to learn about operational restrictions of the application. For example, if we use GPS technology or WiFi to estimate location, it is known that there are many factors that affect the quality and accuracy of the information generated, such as atmospheric conditions and buildings for GPS, or obstacles that affect the WiFi signal, knowledge of these weaknesses shows us situations that may arise due to original uncertainty. On the other hand, if we think of a location-aware application that displays specific information of the place where the user is located, the derivative uncertainty can occur when the distance between two consecutive locations is more than a certain threshold, because a person cannot walk a long distance in a short period of time.

Uncertainty measurement. Consists on establishing the way in which uncertainty is represented, and finding a mechanism to make it tangible. This requires establishing rules to address each of the factors that generate uncertainty identified in the previous phase. The rules follow this structure: IF *condition* THEN *uncertainty activates*, where *condition* is a judgment that involves the uncertainty generation factors and returns a boolean. For example, a rule for GPS technology could be as follows: IF SignalQuality = Low OR SignalQuality = Null THEN uncertainty activates; for WiFi a rule could be: IF PersonsNumber > 10 THEN uncertainty activates; for the case of distance traveled, IF Distance between current location and previous Location > 5 meters THEN uncertainty activates.

Uncertainty treatment. It consists on establishing the actions for dealing with uncertainty. These actions will be executed when one of the rules defined in the previous phase is triggered. The actions are classified into two groups: (1) Automatic handling of uncertainty, is when the application performs activities without user intervention. These actions include: (a) re-estimation, which consists of obtaining the context again; confirmation of context is one of the reasons for this action, another reason is the execution of corrective actions in the source, of course this happens only in situations which enable interaction with the source. A disadvantage of this action is the time and effort required for the re-estimation, and might not be feasible

in applications that need the contextual information immediately. (b) Use of additional sources of information that could be obtained directly without increasing the application complexity. (2) User-assisted uncertainty handling, in these actions the user is aware of the presence of uncertainty. It includes: (a) Requesting feedback, this action informs the user of the existence of uncertainty and requests confirmation of facts or additional information. (b) Informing the user, tells the user of the existence of uncertainty and he is responsible for the decision taken knowing the presence of uncertainty in the received information.

3 Case Study: Location-Aware Electronic Guide for Alejandro Rangel Hidalgo Museum

The Alejandro Rangel Museum at the University of Colima is a space created to disseminate Alejandro Rangel's artistic work, including original paintings, sketches, designs, instruments, furniture and objects used or created by the author. In addition, pre-hispanic pieces from the region are on display.

3.1 Description of the Location-Aware Application

The interactive museum guide shows visitors information related to the pieces on display; images and audio descriptions are included. The application provides specialized information usually given by guides during their tours. The purpose is to harness the technology to facilitate museum staff work and provide an adequate service and attention to visitors. The system is location-aware; it runs on a personal digital assistant device (PDA) and automatically displays information about the piece that the user faces without requiring his direct intervention. The user interface is designed to be simple and easy to use for the visitor not requiring any previous experience with a similar system to use it. The application shows the image and description of the piece; tapping on an image, expands it to provide additional detail; the navigation buttons are used to move to the previous or the next piece; the button "Voz" activates the audio with a description of the piece. The application is shown in Figure 1.

3.2 Dealing with Uncertainty in Location Estimation

Following the uncertainty management process proposed in section 2.2 we studied the context generation mechanism to identify possible sources of original uncertainty.

The location estimation algorithm is based on a back propagation neural network (BP-NN) [21]. The BP-NN receives as input the strength of the access point signal to the wireless network; these values are transformed by the network to an output value that corresponds to user location. However, the signal strength varies significantly, due mainly to rebounds and obstructions of radio frequency waves generated by obstacles present in the physical environment where the wireless network is installed, such as persons or furniture in the building. This can cause the BP-NN to make incorrect estimations. Based on this, we identified the possible presence of original uncertainty. On the other hand, the application uses location to determine the information to display. Thus, if the location is incorrectly estimated the user will receive information related to the wrong piece. This is associated to the presence of derivative uncertainty.

The next step in the procedure is to create rules for uncertainty activation. In the previous phase we identified that original uncertainty can be caused by the number of people present in the room. However, to get this information, additional technology is required and at the same time, the definition of an adequate limit to the number of persons present is needed, increasing the application complexity and introducing new sources of uncertainty. Thus, we decided not to deal with original uncertainty and focus on derivative uncertainty. For this case, we identify the presence of uncertainty when an estimate is different from the previous one. The rule is defined as: IF CurrentEstimate \neq PreviousEstimate THEN Uncertainty.

In the third step of the procedure the actions to be carried out in the presence of uncertainty are established. For this stage we take advantage of an observation made by the museum personnel. They pointed out that visitors normally stop in front of the pieces on a single occasion during their tour, and do not return to observe previously visited pieces. This behavior was used to establish a restricted application for uncertainty treatment, basically re-estimation takes place when the location estimated corresponds to a previously visited location, the purpose of this process is to confirm that the user is now observing a previously visited piece.

As a result of following the three stage procedure we have an uncertainty management heuristic mechanism (UMHM) associated to the location estimation mechanism. This UMHM helps avoid inadequate changes in the information displayed by the application. The general structure of the mechanism is depicted in Figure 1.

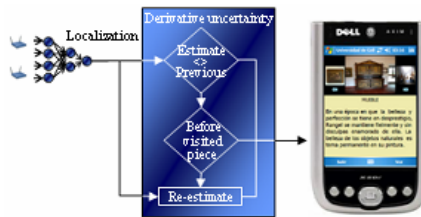


Fig. 1. Uncertainty management heuristic mechanism proposed for location estimation

3.3 Evaluation

The location-aware museum guide was evaluated in two conditions. The first one directly uses the estimation of location information (original application), while the second implements the uncertainty management heuristic mechanism described above (mechanism application).

The purpose of the experiment was to assess the utility of introducing uncertainty management mechanisms in a location-aware application. To this end a utility function was defined including the following elements: (1) the number of times that the information displayed to the user is erroneous according to his location; (2) the number of user interactions with the application; (3) the ease of use of the; and, (4) system reliability. Two experiments were designed to measure these variables.

The first experiment was conducted with 100 visitors in the courtroom's museum following a predefined route. Half the users (50) conducted the tour using the *original application* while the other 50 used the *mechanism application*. They toured the area,

stopping at every piece, looked at the information displayed on the device, and if the information corresponded to the piece he is in front of, he taps on the option "Correct", otherwise he uses the arrow keys to locate the correct information. The application automatically records the number of mistakes and user interactions with the system.

The second experiment involved 118 randomly selected visitors, 60 used the *original application* and 58 the *mechanism application*. The participants were given 30 minutes to complete their tour in no specific order. They were informed of the purpose of the experiment and the general concept of a location-aware application. Then they were given a demonstration of the functionality of the application. They were asked to conduct a tour in total freedom. At the end of tour participants completed a questionnaire containing Likert-type questions.

3.4 Results and Discussion

Based on the elements considered in the definition of the utility function, we established the following hypotheses:

- H1. The application with the UMHM presents the user less erroneous information than the original application.
- H2. The application with the UMHM requires less user interventions than the original application.
- H3. The application with the UMHM is easier to use than the original application.
- H4. The application with the UMHM generates increased user perception of confidence towards the system than the original application.

We used the non-parametric Mann-Whitney test to evaluate the four hypotheses, because the data collected does not come from a normal distribution.

To measure H1 and H2 we used the 100 files collected from the first experiment, these files contain the number of errors in the information displayed and number of interactions of users with the application. Table 1 shows the general statistics of the data. Both hypotheses are accepted since there is a statistically significant difference with both conditions. Thus, using the uncertainty management mechanism reduced the number of erroneous information displayed to the user ($p < 0.00001$) as well as the number of interactions required by the users to operate the system ($p < 0.00001$).

Table 1. General statistics from files collected in the first task

	H1. Number of errors		H2. Number of interactions	
	μ	σ	μ	σ
<i>Original</i>	2.7	0.93	3.4	2.7
<i>Mechanism</i>	1.2	0.94	1.3	1.2

To measure H3 and H4 we used the data from 118 questionnaires collected from the second experiment. The question used to evaluate H3 and H4 are shown in Table 2. The results indicate that H3 is rejected. This means that there is no evidence of difference in the perception of ease of use between the two applications ($p = 0.1565$). On the other hand, H4 is accepted. In other words, the application with the UMHM was perceived to be more reliable by the participants ($p < 0.00001$).

Table 2. Responses to question designed to assess hypotheses H3 and H4

What's your impression about of The "context-aware electronic museum guide"?							
	1	2	3	4	5	6	7
Easy of use.	Easy						Difficult
Original	40 (67%)	13 (21%)	3 (5%)	2 (3%)	1 (2%)	1 (2%)	0
Mechanism	44 (76%)	14 (24%)	0	0	0	0	0
Reliability.	1	2	3	4	5	6	7
	Reliable						Not reliable
Original	9 (15%)	17 (28%)	19 (32%)	8 (14%)	5 (8%)	2 (3%)	0
Mechanism	31 (53%)	17 (29%)	8 (14%)	1 (2%)	1 (2%)	0	0

4 Related Work

Several papers have proposed solutions to deal with context uncertainty; these can be classified in quantitative approaches aimed at estimating uncertainty and those which involve user participation. Among the former, Tau Gu *et al.* [4] used Bayesian networks to estimate the activity of a user. They use ontologies to define context, whose definition includes probability values and relationship links. Truong *et al.* [6] also used Bayesian networks and ontologies, but their focus is on reusing context ontology definitions. They outline an ontology structure that may be used in different scenarios. Ranganathan *et al.* [5] proposed probabilistic reasoning and fuzzy logic to deal with uncertainty. They represent contexts as predicates, following the convention that the predicate's name is the type of context being described (such as location, temperature, or time). They model uncertainty by attaching a confidence value between 0 and 1 to predicates. This value measures the probability (in the case of probabilistic approaches) or the membership value (in the case of fuzzy logic) of the event corresponding to the context predicate being true. Guan *et al.* [9] also use fuzzy logic for handling uncertainty in high-level context inference, the proposal is based on the use of a fuzzy decision tree to establish rules that define contexts. However, these proposals are based on quantitative methods for uncertainty treatment, which require a great amount of additional work and necessary information collection for its implementation in the real world, and in some situations it is very difficult or impossible to obtain such information. Besides, none of these proposals establish how the application should behave in the presence of uncertainty, rather they focus on the process of uncertainty identification.

Another set of proposals involve user intervention. Xu & Cheung [7] defined uncertainty as context inconsistencies, they present a proposal that lets users define context patterns which can generate uncertainty. The basic idea is to identify situations that can generate context contradictions and implement actions to correct them. The proposal requires the definition of contradictory context patterns, which in many situations can be a rather complex task. Other proposals are based on the direct

intervention of the user to manage uncertainty. For instance, Dey et al. [22] define mediation as a dialog between a human and a computer that resolves ambiguity, mediation can conceptually be applied whenever misunderstandings arise between application and user. In this sense, Antifakos et al. [23] propose a strategy in which the uncertainty in the application, is notified to the user, to help him decide the appropriate course of action.

5 Conclusions

Uncertainty in context-aware applications can be generated by the presence of uncertain, ambiguous or incorrect contextual information. Several proposals have been made for dealing with context uncertainty; these can be classified in quantitative approaches aimed at estimating uncertainty and those which involve the user. However, no empirical evidence has been provided regarding problems associated with uncertainty management in context-aware applications. In this document we establish a classification of uncertainty that separates the acquisition and processing of context, as well as a three-stage procedure for the creation of an uncertainty management heuristic mechanism. The use of this procedure is illustrated with the implementation of a location-aware interactive museum guide, which was used in an experimental evaluation to measure the impact and utility of the approach.

The results of the experiment provide empirical evidence that the uncertainty is a factor that affects the reliability of a context-aware application. This can cause the user to cease using the application if he questions the reliability of the information it provides. They welcomed the use of this technology during their visit to the museum. In addition, they found the context-aware application to be useful and easy to use.

References

1. Raptis, D., Tselios, N., Avouris, N.: Context-based design of mobile applications for museums: a survey of existing practices. In: 7th international conference on Human computer interaction with mobile devices & services (MobileHCI 2005). ACM, Salzburg (2005)
2. Satyanarayanan, M.: Pervasive Computing: Vision and Challenges. *IEEE Personal Communications* 8, 10–17 (2001)
3. Satyanarayanan, M.: Coping with Uncertainty. *IEEE Pervasive Computing* 2, 2 (2003)
4. Gu, T., Pung, H.K., Zhang, D.Q.: A Bayesian Approach for Dealing with Uncertain Contexts. In: 2nd Intl. Conf. on Pervasive Computing (Pervasive 2004), Austria, vol. 176 (2004)
5. Ranganathan, A., Al-Muhtadi, J., Campbell, R.H.: Reasoning about Uncertain Contexts in Pervasive Computing Environments. *IEEE Pervasive Computing Journal* 3, 62–70 (2004)
6. Truong, B.A., Lee, Y.-K., Lee, S.-Y.: Modeling and Reasoning about Uncertainty in Context-Aware Systems. In: IEEE International Conference on e-Business Engineering (ICEBE 2005), pp. 102–109. IEEE Computer Society, Beijing (2005)
7. Xu, C., Cheung, S.C.: Inconsistency detection and resolution for context-aware middleware support. In: 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering ESEC/FSE-13, vol. 30, pp. 336–345. ACM Press, Lisbon (2005)

8. Dey, A.K., Mankoff, J., Abowd, G.D., Carter, S.: Distributed mediation of ambiguous context in aware environments. In: Proceedings of the 15th annual ACM symposium on User interface software and technology, pp. 121–130. ACM Press, Paris (2002)
9. Guan, D., Yuan, W., Gavrilov, A., Lee, S., Lee, Y.-K., Han, S.: Using Fuzzy Decision Tree to Handle Uncertainty in Context Deduction. In: Huang, D.-S., Li, K., Irwin, G.W. (eds.) ICIC 2006. LNCS (LNAI), vol. 4114, pp. 63–72. Springer, Heidelberg (2006)
10. RAE: Página de la Real Academia Española. Real Academia Española, España (2008)
11. Penrod, J.: Living with uncertainty: concept advancement. *Journal of Advanced Nursing* 57, 658–667 (2007)
12. Li, D., Du, Y.: *Artificial Intelligence With Uncertainty*. Chapman & Hall/CRC, Taylor & Francis Group (2008)
13. Truong, B.A., Lee, Y.-K., Lee, S.-Y.: A unified context model: Bringing probabilistic models to context ontology. In: Enokido, T., Yan, L., Xiao, B., Kim, D.Y., Dai, Y.-S., Yang, L.T. (eds.) EUC-WS 2005. LNCS, vol. 3823, pp. 566–575. Springer, Heidelberg (2005)
14. Korpipää, P., Mäntyjärvi, J., Kela, J., Keränen, H., Malm, E.-J.: Managing Context Information in Mobile Devices. *IEEE Pervasive Computing Journal* 2, 42–51 (2003)
15. Benford, S., Crabtree, A., Flinham, M., Drozd, A., Anastasi, R., Paxton, M., Tandavanitj, N., Adams, M., Row-Farr, J.: Can you see me now? *ACM Transactions on Computer-Human Interaction* 13, 100–133 (2006)
16. Xu, C., Cheung, S.C., Chan, W.K.: Incremental Consistency Checking for Pervasive Context. In: 28th International Conference on Software Engineering (ICSE 2006), Shanghai, China, pp. 292–301 (2006)
17. Beeharee, A., Steed, A.: Exploiting real world knowledge in ubiquitous applications. *Personal and Ubiquitous Computing* 11, 429–437 (2007)
18. Poole, D., Smyth, C.: Type Uncertainty in Ontologically-Grounded Qualitative Probabilistic Matching. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 763–774. Springer, Heidelberg (2005)
19. Lachapelle, G.: Pedestrian navigation with high sensitivity GPS receivers and MEMS. *Personal and Ubiquitous Computing* 11, 481–488 (2007)
20. Indulska, J., Sutton, P.: Location management in pervasive systems. In: The Australasian information security workshop, vol. 34, pp. 143–151. ACM, Adelaide (2003)
21. Castro, L.A., Favela, J.: Continuous Tracking of User Location in WLANs Using Recurrent Neural Networks. In: Sexto Encuentro Internacional de Computación (ENC 2005), pp. 174–181. IEEE Press, Puebla (2005)
22. Dey, A.K., Mankoff, J.: Designing Mediation for Context-Aware Applications. *ACM Transactions on Computer-Human Interaction* 12, 53–80 (2005)
23. Antifakos, S., Kern, N., Schiele, B., Schwaninger, A.: Towards Improving Trust in Context-Aware Systems by Displaying System Confidence. In: 7th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 2005), vol. 111, pp. 9–14. ACM, Salzburg (2005)

Modeling Context Life Cycle for Building Smarter Applications in Ubiquitous Computing Environments

Hyunjun Chang, Seokkyoo Shin, and Changshin Chung

Telecommunications Technology Association (TTA)
267-2 Seoheon-dong, Bundang-gu, Seongnam-city, Gyeonggi-do, Korea
{hjchang, skshin, cschung}@tta.or.kr

Abstract. In this paper, we propose a new scheme for modeling context life cycle in order to build smarter applications in ubiquitous computing environments. Existing context management system implicitly assumes that a context representing user's situation can be fallen into just one of the two states such as valid or invalid. We observed that, however, a context can have additional state and transfer from one state to another seamlessly according to the change of a user-related context such as user's location. The behavior of corresponding applications can also be finely tuned so that we can provide more adaptive services. For demonstration, we use the ontology for modeling and enriching the life cycle of a context. It enables us to utilize high-level inference capability which derives the current situation of a user. We also enhance the functionality of existing context management middleware using the context life cycle management scheme in this paper.

Keywords: Context Life Cycle, Context Modeling, Context-Awareness.

1 Introduction

Ubiquitous computing makes it possible to build smart spaces that allow users to exploit relevant information and services based on their situations recognized through various context information [3, 6, 14]. In the smart spaces, users do not have to take explicit action for provision of the needed services. Hence users can keep concentrating on their current activities without distraction to manually configure their computing environment [6].

Existing context management researches such as CoBrA, SOCAM, Aura, and Gaia use a management policy to guide the system behavior in different contexts [2, 5, 6, 14]. A policy is often described by an Event-Condition-Action (ECA) rule such as 'on *Event* if *Condition* then *Action*'. It means when an event occurs and the specified contextual conditions are satisfied then the corresponding actions are carried out. For example, if a user changes his/her location to a shower booth (an event), and a 'take shower' context is recognized (a condition), then a heater service adjusts the water temperature preferred by the user (an action). They implicitly assume that the context can be fallen into only one of the two states such as valid or invalid.

Each context, however, activates services for the user and is subject to certain duration of validity, which we call a *context life cycle*. In addition, it is possible for

multiple contexts to be active in the common time frame, which allows some other contexts may terminate, temporarily pause or even coact with contexts depending on the relationships among them. During its lifetime, a context has more than one state and moves from one state to another. For example, when a user changes his/her location out of the shower booth, the active ‘take shower’ context could be changed into an inactive (i.e., terminated) state due to the location change event. In contrast, if users in a ‘attend a meeting’ context leave the meeting room for break, the ‘attend a meeting’ context had better be paused, but not terminated, until the user comes back in order to continue the meeting activity seamlessly. The corresponding application services are also required to adapt themselves according to the state of a relevant context during its life cycle.

H. Chen et al. [3] proposed a reasoning scheme to deal with an inconsistent context in a smart meeting room environment. The scheme keeps the active context state unchanged by nullifying the unexpected user’s location change through both default reasoning and abductive reasoning. S. Urbanski et al. [13] proposed a process-based model for defining applications as a sequence of steps using a specially designed language called Sentient Process Command Language (SPCL). Each step allows application developers to specify the condition of various states of the step such as activation, pause, finish, and abort to let the system know about the state change during the execution of the step. However, both of them did not take it into account how services should be adapted to the state transition of the context.

In this paper, we propose a new context life cycle management scheme based on an extended ECA policy, called Event-Condition-State-Action (ECSA) policy so that application developers can specify how service behaviors should be adapted to the state transition of a context in the policy structure. We also present the context relationship and the runtime system architecture which makes use of it for the system to capture the state transition of a context. We implemented our proposed scheme based on our ubiquitous computing middleware system.

This paper is organized as follows. The concept of context life cycle is described in Section 2. The context life cycle management scheme and the ECSA policy are discussed in Section 3. Its implementation is presented in Section 4. Finally, the conclusion and future work follow in Section 5.

2 Modeling Context Life Cycle

A context can be regarded as a syntactic interpretation of the states of the present environment where some services are in action. Some contexts are generated and terminated instantly by themselves such as ‘Jane enters home’ that is recognized by the sensors at the front door. On the contrary, other contexts are generated and remain active for a relatively long duration so as to get associated with each other in the same space such as ‘Jane is watching TV while cooking’. The former is often called an instant event and the latter called an interval event [1, 10]. In this paper, we define all contexts have their respective durations of validity no matter whether instant or interval. We call it context’s life cycle.

In the management of life cycle of contexts, multiple contexts may take place in the common time frame and some contexts may get paused or terminated by another

context. For example, when Jane moves to the kitchen and turns on the cooking stove, the ‘cooking’ context is in an *active* state. When Jane leaves the kitchen after turning off the stove, the ‘cooking’ context changes into an *inactive* state. On the contrary, Jane may do more than one activity in a concurrent manner, such that if Jane wants to ‘answer telephone’ while she is ‘cooking’, she moves to living room where the telephone is located. Then, the state of the previous ‘cooking’ context should not be terminated but *paused* temporarily and *resumed* when she returns to cook. A context can be designed to set another context instead of just terminating it.

In consequence, we define the life cycle of a context has one of the three possible states: *Active*, *Inactive* and *Paused*, which will form six possible state transitions as the basic state transition modeling metric. It is noted that resumption after a pause can be described as another active state.

- *Active*

A context is said to be in an *active* state and its corresponding service(s) is activated for a certain duration. A context becomes active as soon as an event generates a context and service(s) is activated. For example, sensed events in the bathroom lead to generation of a context, ‘take shower’ and its corresponding services such as heater and ventilator start working. The context is terminated by new contexts and enters into an *inactive* state.

- *Inactive*

A context is in an *inactive* state when it is not valid anymore because the conditions of the context turn into false or it cannot coexist with another active context. It is fair to say that all contexts are in the inactive states by default before their life cycle’s begin. Likewise, a context life cycle begins when it becomes active from inactive and ends when the context becomes back to inactive state.

- *Paused*

With some new contexts’ involvement, a context enters in a *paused* state when it requires not to get terminated at all but, instead, to maintain some modified services. The paused state changes to another active state to resume the previous service.

The context life cycle is made up with some possible combination of the above-stated primitive states in terms of the state transition. Among them only five transitions are possible except the paused state transitioning from inactive state.

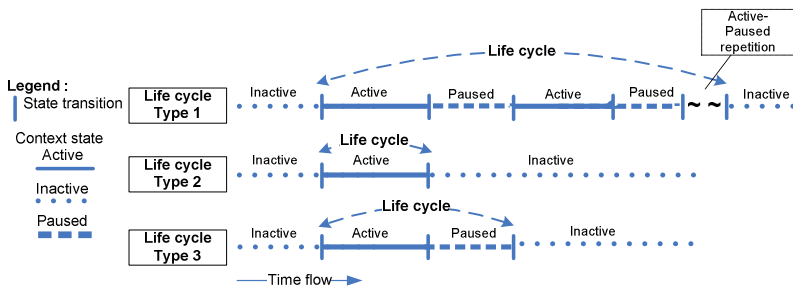


Fig. 1. Types of context life cycle

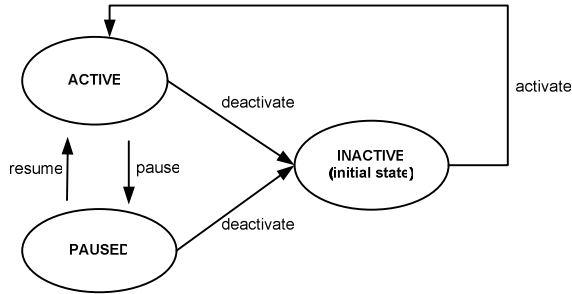


Fig. 2. Context state transition

Based on the possible combinations, a context life cycle falls into one of the three types as shown in Fig. 1. Type 1 describes that paused states and active states are in a life cycle. Type 2 describes an active state changed into the inactive state. Type 3 shows a life cycle with an active state and a paused state which become terminated without a following active state.

Fig. 2 shows the context state transition diagram. The diagram includes four types of state transition events which change a context state from one state to another. Events are usually captured by context life cycle manager monitoring the states of all valid contexts. The context state change is made by four triggering events: activate, pause, resume and deactivate, and services are provided according to various relationships among contexts and their service availabilities which are identified and designed by application service developers. The detailed context relationships are discussed in Section 3.2.

3 Building Smarter Applications Based on Extended ECA Policy

In this section, we discuss the components of context life cycle management to support multiple contexts seamlessly. We employ a new ECSA policy and context relationships. The ECSA policy allows applications developers or users to specify how they customize service behaviors along with the life cycle of a context.

3.1 ECSA Policy

The services initiated by the active context needs to adapt themselves along with the state transition of the context until the context is terminated. The proposed ECSA policy has been designed with four state transition events that enable application developers to specify how the services should adapt themselves to each context state transition as described in Fig. 3. Whenever the context state is changed, context life cycle manager is notified with the StateChanged event and responsible for enforcing all the prescribed policies to execute the service action. The changed context state is delivered as events and policy actions are triggered accordingly. The ECSA policy has the following form in case of the ‘take shower’ context in a smart home which contains audio, heater, and light service.

Application developers are able to benefit from the ECSA policy creating a new policy and updating existing policies for a context. We could define individual policies for each state in a separate manner. However, it does not guarantee the system to work reliably unless the application developers design individual policy for each state of a context. Moreover, any missing policy for a certain state of a context will lead to inconsistent service behaviors since policy actions for each state of the context are tightly coupled with one another.

```

on StateChanged(Context c) {
  when (c.getSituation() == 'take shower')
  activate :
    do {PlayMusic(), On(WaterHeater), On(Light)}
  pause :
    do {PauseMusic(), On(WaterHeater), Dim(Light)}
  resume :
    do {ResumeMusic() ,On(WaterHeater), DimUp(Light)}
  deactivate :
    do {StopMusic(), Off(WaterHeater), Off(Light)}
}

```

Fig. 3. ECSA policy for 'take shower' context in home

For example, the policy actions of the *deactivate* should nullify all the effects of the policy actions of the *activate* in order to maintain the consistency of the system states. The policy actions of the *resume* should do so against the policy actions of the *pause*. These functional dependencies can be overlooked by application developers unless the context life cycle is managed in such a centralized manner. Like ECA a policy can also be defined on an event such as a user's location change. In this case, the ECSA policy enables application developers to specify actions only on the *activate* and leave the others empty.

3.2 Context Relationship

When more than one context are active at the same time, the subsequent context might terminate, pause, or coexist against the previous context. We exploit the context relationships to coordinate the context states. The relationships are classified into three types: contradictory, mutually adaptive and concurrent.

The *contradictory* relationship implies the subsequent context terminates the previous context. For example, when Jane 'takes shower' and changes her location to bedroom for 'sleeping,' the former context is immediately deactivated when the system recognizes she has 'sleeping' context assuming that 'sleeping' context does not go with other contexts concurrently.

In a *mutually adaptive* relationship, the previous context becomes paused when the subsequent context is activated. For example, when Jane 'takes shower,' the context is paused while a certain interrupting context such as 'telephone ringing' is active and she changes location for a short time to answer the phone. After the 'telephone

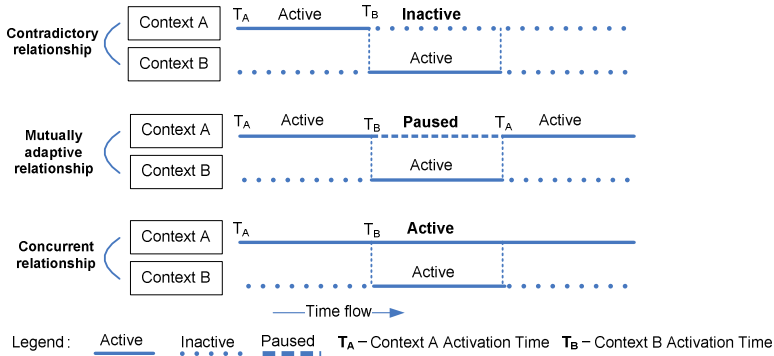


Fig. 4. Types of context relationships

ringing’ is deactivated and she returns, the former ‘take shower’ context resumes its state so that the system will resume its service by resuming music from the point of time she left out. This can be realized using the ECSA policy described in Fig. 3 in Section 3.1.

In a *concurrent* relationship, the previous context may coexist with the following context thus services are in action concurrently. For example, if Jane wants ‘watching TV’ while she is ‘cooking’, the previous ‘cooking’ context should be remain in an active state when the ‘watching TV’ context is activated. She is able to get the two separate services at the same time. Fig. 4 illustrates the three types of relationships and the corresponding context states.

Context Lifecycle Manager (CLM) works between Context Manager (CM) and Context-aware Policy Manager (CPM) as shown in Fig. 5. A new context is delivered from CM to CLM. Then, the CLM refers to the context relationship to determine what contexts need to be paused or deactivated due to the new context. All existing contexts with its current state are contained in the context database. CLM updates the state of each context of the context database according to the relationship, and then, the StateChanged event is dispatched to CPM which triggers application services as specified in the ECSA policy for each StateChanged event. Consequently, CPM is

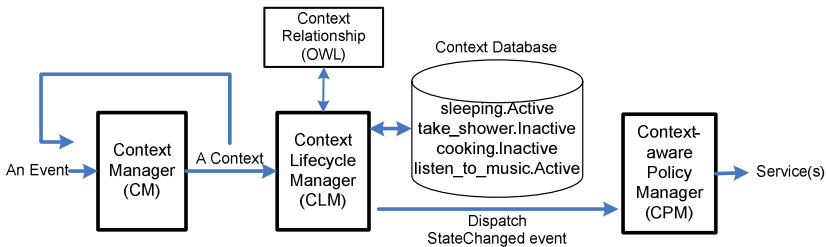


Fig. 5. Enhanced context-aware service(s) provision with the context life cycle manager

notified with not only the events for new contexts but also the events for paused, resumed i.e., active, and inactive contexts.

Fig. 6 demonstrates an example of a context relationship specification among contexts in a home environment using Web Ontology Language (OWL) [15]. Both ‘take shower’ and ‘cooking’ contexts are contradictory to ‘sleeping’ context. ‘Listen to music’ context may cowork with ‘sleeping’ context concurrently. ‘Cooking’ context is mutually adaptive relationship with ‘answer telephone’ context.

```

<tta:Context rdf:ID = "sleeping">
  <tta:isContradictory rdf:resource = "take_shower"/>
  <tta:isContradictory rdf:resource = "cooking"/>
  <tta:isConcurrentWith rdf:resource = "listen_to_music"/>
</tta:Context>

<tta:Context rdf:ID = "cooking">
  <tta:isMutuallyAdaptive rdf:resource = "answer_telephone"/>
</tta:Context>
    
```

Fig. 6. Context relationship specification using OWL

We have designed an ontology which describes knowledge about context state and the context relationship as given in Fig. 7.

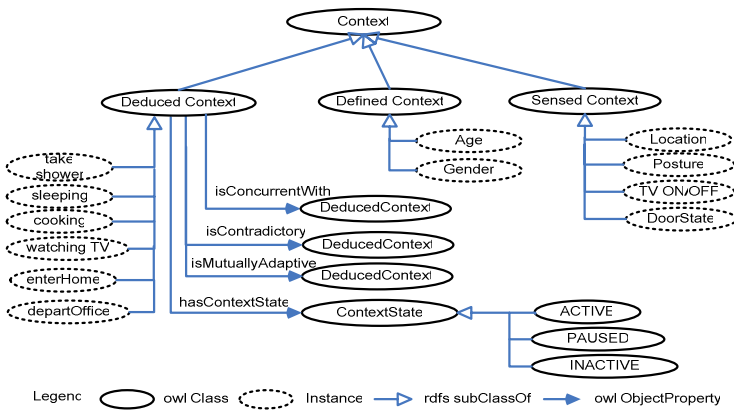


Fig. 7. Graphical representation of the context relationship ontology

Contexts typically consist of the sensed context obtained mainly from the physical sensors, the defined context specified by a user or system administrator, and the deduced context inferred from the former two kinds of contexts by using a context reasoner [5]. The deduced context has predicate ‘isConcurrentWith’ and ‘isContradictory’ to represent the list of contexts that can or cannot coexist with it, respectively. Also, it has predicate ‘hasContextState’ to represent the current state of a context during its life cycle.

4 Implementation

We implemented our scheme on top of the ubiquitous computing middleware system. To build them, we used J2SE 5.0 and Jena [8] semantic web framework, which includes the rule-based inference engine. Fig. 8 shows the system architecture of the Context Manager (CM) and the Context Lifecycle Manager (CLM). The left part of the block diagram depicts the middleware components for the context-awareness, which is inspired by the Context Toolkit [4]. It is used in gathering, aggregating, inferring, and disseminating contexts.

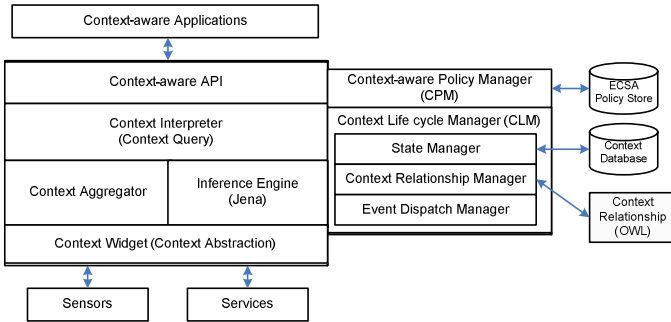


Fig. 8. System architecture for the context-awareness and context life cycle management

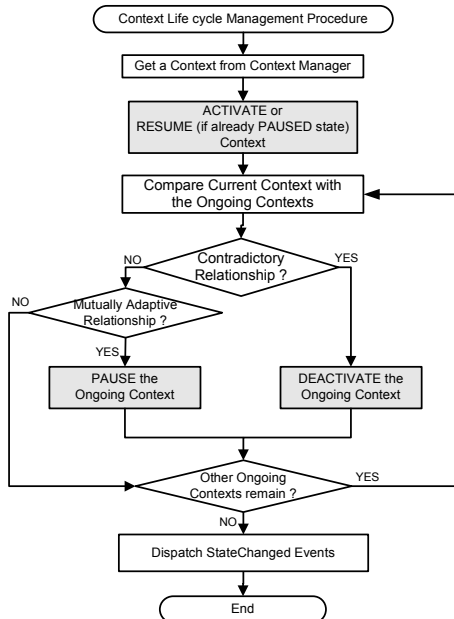


Fig. 9. ECSA-based context life cycle management procedure

On top of the middleware, the CLM coordinates the life cycle of all contexts. It consists of the three subcomponents, State Manager, Context Relationship Manager, and Event Dispatch Manager as shown on the right side of the diagram. State Manager monitors states of all the contexts existing in the context database. Context Relationship Manager makes use of the context relationships to determine the state of the context intervened by another context. Event Dispatch Manager forwards state transition events to the Context-aware Policy Manager.

The CLM executes the life cycle management procedure described in a flowchart in Fig. 9. Whenever a new context is delivered into the CLM, the CLM inquires whether the delivered context has been one of the paused context at first, which means that the context has been paused due to the intervention of another context and then is to be resumed. CLM compares the active context i.e., newly activated or resumed with the ongoing (i.e., active or paused) contexts one by one and updates the states of the ongoing contexts. The context relationships are referred in this step. For each state transition which is shown as gray boxes in Fig. 9, a StateChanged event is generated and notified to the Context-aware Policy Manager.

5 Conclusion and Future Work

The existing context management methodologies based on the ECA policy take it into account only what actions need to be executed when the contextual condition is satisfied. It does not concern the termination of the executed actions. This may cause system inefficiency such as wasting water and electricity in a home environment. Another ECA policy for taking an opposite actions could avoid such inefficiency, but it is error-prone because application developers should consider all such potential problematic cases whenever he/she generates new policies. In this paper, we thus proposed a context life cycle management scheme based on the ECSA policy for more adaptive action execution according to the changing context states. Four types of transition events such as activate, pause, resume, and deactivate are generated by the system when a context changes its state into another. The ECSA policy exploits the four different types of events in specifying the corresponding service behaviors. This enables the ubiquitous computing systems to have the fine-grained adaptability to the user's changing context. As a future work, we plan to measure the degree of user's satisfaction through the user study in the real smart home environments based on our ubiquitous computing middleware system.

References

1. Allen, J.F.: Actions and Events in Interval Temporal Logic. *Journal of Logic and Computation* 4(5), 531–579 (1994)
2. Chen, H., Finin, T., Joshi, A.: Semantic Web in the Context Broker Architecture. In: *Proceedings of the 2nd Annual IEEE International Conference on Pervasive Computer and Communications*, p. 277 (2004)
3. Chen, H., Perich, F., Chakraborty, D., Finin, T., Joshi, A.: Intelligent Agents Meet Semantic Web in a Smart Meeting Room. In: *Proceedings of the 3rd International Joint Conference on Autonomous Agents & Multi Agent Systems (AAMAS 2004)*, pp. 854–861 (2004)

4. Dey, A.K., Abowd, G.D., Salber, D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *HCI Journal* 16(2-4), 97–166 (2001)
5. Gu, T., Pung, H.K., Zhang, D.Q.: A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications* 28(1), 1–18 (2005)
6. Garlan, D., Siewiorek, D., Smailagic, A., Steenkiste, P.: Project Aura: Toward Distraction-Free Pervasive Computing. *IEEE Pervasive Computing special issue on Integrated Pervasive Computing Environments* 21(2), 22–31 (2002)
7. Hyunjun, C., Seokkyoo, S., Changshin, C.: Context Life Cycle Management Scheme in Ubiquitous Computing Environments. In: *The 2nd IEEE International Workshop on Managing Context Information and Semantics in Mobile Environments (MCISME) held in conjunction with the 8th IEEE International Conference on Mobile Data Management, MDM 2007* (2007)
8. Jena Semantic Web Framework, <http://jena.sourceforge.net>
9. MacKworth, A.K., Goebel, R.G., Poole, D.I.: *Computational Intelligence: A Logical Approach*, pp. 319–342. Oxford Univ. Press, Oxford (1998)
10. Pan, F., Hobbs, J.R.: Time in OWL-S. In: *Proceedings of AAAI 2004 Spring Symposium on Semantic Web* (2004)
11. Shankar, C., Campbell, R.: A Policy-based Management Framework for Pervasive Systems using Axiomatized Rule Actions. In: *4th IEEE International Symposium on Network Computing and Applications (IEEE NCA 2005)*, pp. 255–258 (2005)
12. Shankar, C., Campbell, R.: Ordering Management Actions in Pervasive Systems using Specification-enhanced Policies. In: *4th IEEE International Conference on Pervasive Computing and Communications*, pp. 234–238 (2006)
13. Urbanski, S., Becker, C., Rothermel, K.: Sentient Processes – Process-based Applications in Pervasive Computing. In: *Work in Progress Reports of the 4th Annual IEEE International Conference on Pervasive Computing and Communications*, pp. 608–611 (2006)
14. Wang, X., Dong, J.S., Chin, C.Y., Hettiarachchi, S.R., Zhang, D.: Semantic Space: An Infrastructure for Smart Spaces. *IEEE Pervasive Computing* 3(3), 32–39 (2004)
15. Web Ontology Language (OWL) (2004), <http://www.w3.org/2004/OWL/>

Game Development Framework Based Upon Sensors and Actuators

Ray van Brandenburg, Arie Horst, Bas Burgers, and Nirvana Meratnia

Department of Computer Science, University of Twente, Enschede, The Netherlands
{r.vanbrandenburg,a.p.horst,b.j.burgers}@student.utwente.nl,
n.meratnia@ewi.utwente.nl

Abstract. Urge for comfort and excitement have made gadgets indispensable part of our life. The technology-enabled gadgets not only facilitate and enrich our daily lives but also are interesting tools to challenge human imagination to design and implement new ubiquitous applications. Pervasive gaming, in which human interaction and game/scenario-dependent designs are often common practices, has proved to be one of the areas to successfully combine technology and the human fantasy. By moving away from games being played by humans and by focusing instead on games played by robots and giving humans the leading role of defining game strategies and players' roles, this paper aims at bridging the two fields of robotics and wireless sensor/actuator networks and exploring their potentials in the field of pervasive gaming. A generic game development framework is introduced that accommodates different types of robots and various kinds of sensors and actuators. Being extensible and modular, the proposed framework can be used for a wide range of pervasive applications built upon sensors and actuators. To enable game development, a Wiimote-based robot identification and localization technique is presented. The proposed framework and robot identification, localization, control and communication mechanisms are evaluated by implementing a game example.

1 Introduction

With no doubt technology has changed the way we perceive the world, communicate and interact with others and our environment, live our lives, perform our jobs and entertain ourselves. The great ubiquitous computing vision of Mark Weiser is not far from our reach and researchers and developers have already started challenging the limits of this vision by bringing technologies and human imaginations to the next level. Today more than ever, gadgets “recede into the background of our lives” [1]. The technology-enabled gadgets not only facilitate and enrich our daily lives and increase productivity at work but also are interesting tools to challenge the humans imagination to design and implement new ubiquitous applications. Pervasive gaming has proved to be one of the areas to successfully combine technology and the human fantasy. Schneider et al. define pervasive game as “live-action roleplaying game that is augmented with computing and communication technology in a way that combines the physical and digital space together” [2]. Although this definition does not explicitly specifies players to be humans, the majority of designed pervasive games are centered around human players and have a strong focus on human interaction [3,4,5,6].

In outdoor game scenarios, human identification and localization is often based on large-scale localization using GPS or GSM [7]. Indoor localization on the other hand is generally based on beacon triangulation [8,9,10]. The accuracy of both indoor and outdoor localization techniques significantly varies depending on the technology and the environment where the technology is used and it can be anywhere between few meters to few centimeters.

By moving away from games being played by humans and by focusing instead on games played by robots and giving humans the leading role of defining game strategies and players' roles, this paper aims at bridging the two fields of robotics and wireless sensor/actuator networks and exploring their potentials in the field of pervasive gaming.

2 Identification and Localization

One of the most important aspects of almost every pervasive game is the knowledge of own, and possibly the opponents' location. In addition, it is highly useful to be able to distinguish one player from the other. In other words, the majority of pervasive games, if not all, require player identification and localization. Although identification and localization may sound like two different problems, they may both be solved using the same technique. One should recall that players of the pervasive games we have in mind are robots. Not restricting ourselves to robots that have built-in hardware capable of localization, there is a need to use external sensors for localization and identification. The choice of external sensors depends on many factors such as cost, precision, and flexibility, to name but a few. Possible solutions for both identification and localization include:

- **Beacon triangulation:** By placing either infrared or ultrasound beacons on certain positions on the playing field, it is possible to triangulate the location of an object, for example a robot, using an IR detector placed on the object itself. The main advantage of this approach is its low cost. A disadvantage is that every robot should have its own detector. This technique is more suitable for localization than for identification. Moreover, it requires developing a special communication line for the framework to communicate with the IR detector.
- **Digital camera:** Using a camera and image recognition software it is possible to locate and identify different robots, which can be distinguished by for example, different colored patches. This is a method often used in Robosoccer [11]. The advantages are high accuracy and relatively simple communication mechanism. The main drawback may be the complexity of image recognition software.
- **Wiimote [12]:** Wiimote is a game controller used by the Nintendo Wii game console. It contains accelerometers as well as an infrared camera. Communication with the Wiimote is over Bluetooth. Using a stationary Wiimote, it is possible to detect IR LEDs mounted on robots. The main advantages of this approach are its relative simple and fast implementation and reasonably good identification and localization accuracy. The Wiimote is also a cheap

solution compared to the digital camera. Disadvantages include limited precision at larger distances and also that one Wiimote can only 'see' up to a maximum of four LEDs.

2.1 Wiimote

Because of the relatively good identification/localization accuracy and the easy Bluetooth connectivity, we use the Wiimote for robot identification/localization. It can be placed anywhere in the playing field to locate both static and mobile robots equipped with IR LEDs.

In presence of more than four IR sources, Wiimote arbitrarily chooses which LEDs to save in its registry and which ones to discard. The order in which the Wiimote passes the IR source data to the Bluetooth controller is random. In fact the Wiimote does not really track an IR source and rather just passes the positions of up to four IR sources to the recipient. Therefore, constant mapping between incoming IR data and one of the tracked objects is needed.

A 'must have' feature of the framework is the ability to distinguish between multiple robots, for example between friend and foe, or between two members of the same team. Using the Wiimote, there are multiple possible solutions to this problem. The most obvious solution is to let the LEDs blink at different frequencies. However, there are two drawbacks to this approach. First, internal signal processing inside the Wiimote makes determining the exact frequency with which the LEDs blink hard. To solve this problem, it would be necessary to put the different LEDs very far from each other. This in turn would make tracking the blinking LEDs more difficult. A simpler solution is to determine the identity of a robot by the number of LEDs it has on top. For example, one LED means robot 1 (or team one) and two LEDs means robot 2 (or team two). However, because the Wiimote can only track up to four LEDs simultaneously, this approach limits number of robots the Wiimote can track at the same time.

Communication with the Wiimote is done over Bluetooth using an open source C library called WiiUse [13]. Functions of this library include reading IR and accelerometer data from the Wiimote and changing various internal Wiimote parameters such as IR-sensitivity.

2.2 Localization Technique

An important step in the Wiimote-based localization is transforming raw camera coordinates to locations relative to the Wiimote in the 2D playing field. In order to do so, one constraint has to be introduced. This constraint is that all the LEDs need to be at the same height to convert the 3D problem to a 2D problem. Fig 1, in which various variables used in the transformation are shown, presents a side view of a Wiimote and an IR LED source.

The transformation from raw camera coordinates to positions relative to the Wiimote is carried out using the equations (1) and (2), in which ϕ_{FOV} is the field of view of the camera, X_{res} and Y_{res} are the resolution of the camera in x and y, x_{raw} and y_{raw} are the raw camera coordinates and h_{diff} is the difference in height of the LED and the Wiimote. Equations (1) and (2) are a function of, among other things, the pitch of the Wiimote. While it is of course possible to determine this pitch by hand, it is more

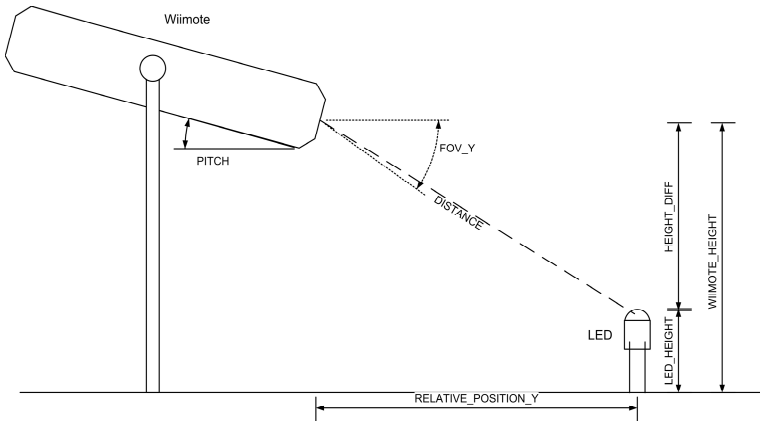


Fig. 1. The side view of the Wiimote

accurate and more dynamic to let the Wiimote itself determine this pitch using the built-in accelerometers. This is done by measuring the direction in which Earth’s gravity points relatively to the three-axis-accelerometer. This way when the Wiimote changes pitch, for example because it is mounted on an actuator, it can re-determine its pitch and recalculate all locations automatically.

$$y_{relative} = h_{diff} \cdot \tan \left(\frac{y_{raw}}{Y_{res} / \varphi_{FOV,Y}} + (90 - \theta_{pitch} - 0.5 \cdot \varphi_{FOV,Y}) \right) \tag{1}$$

$$x_{relative} = \sqrt{(y_{relative})^2 + h_{diff}^2} \cdot \tan \left((0.5 \cdot \varphi_{FOV,X}) - \frac{x_{raw}}{X_{RES} / \varphi_{FOV,X}} \right) \tag{2}$$

$$\begin{bmatrix} x_{world} \\ y_{world} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_{relative} \\ y_{relative} \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} \tag{3}$$

Equations (1) and (2) calculate the position of an IR source relative to the position of the Wiimote. In a game scenario it could be useful to mount the Wiimote on some sort of rotating actuator, which requires determining the positions independently from the Wiimotes own position and orientation. This can be done using the equation (3), in which θ is the rotation of the Wiimote in the (x,y) plane.

2.3 Localization Precision

Several measurements have been performed to test the precision of localization using the Wiimote. Data from these measurements is shown in Fig 2. As it can be seen, the precision of the Wiimote-based localization in the x-direction is fairly constant; an average error of about -10mm to 10mm. In the y-direction, however, the error

strongly depends on the distance to the object. This error in the y-direction can be split into three distinct areas, i.e., short, medium and large distance.

At very short distances from the Wiimote, error is relatively large (about 40mm). The reason is twofold. First, when the LED is very close to (beneath of) the Wiimote, only just within the Wiimotes vertical field of view, the accuracy of the IR camera is not very good. This effect can also be seen in the figure depicting the error in the x-direction, where the error is the largest at the places where the Wiimote is only just visible (at -500mm and 500mm). This could be due to the lens distortion effect. Secondly, at short distances from the Wiimote, the error resulting from the pitch calculation has a larger effect and leads to a larger error in position determination.

At medium distances, the Wiimote performs quite well, having an average error of about 12mm. This error is comparable to the error found in the x-direction.

At large distances, the limited resolution of the IR camera together with the fact that the height of the Wiimote compared to the distance to the object is small, result in a lower accuracy and an average error of about 35mm.

The maximum range of the Wiimote highly depends on the specific type of LED used. In our experiments performed with a small standard LEDs, the maximum range appeared to be around 2 meters when the LED was pointed directly at the Wiimote. So-called ultra-bright LEDs could dramatically increase this range.

The accuracy of the proposed technique is good for distances up to three meters and is comparable with infrared beacon triangulation methods such as the one described in [8].

One possible way to increase localization accuracy is to use two Wiimotes and to combine the data to form an image of the playing field with better precision. Another

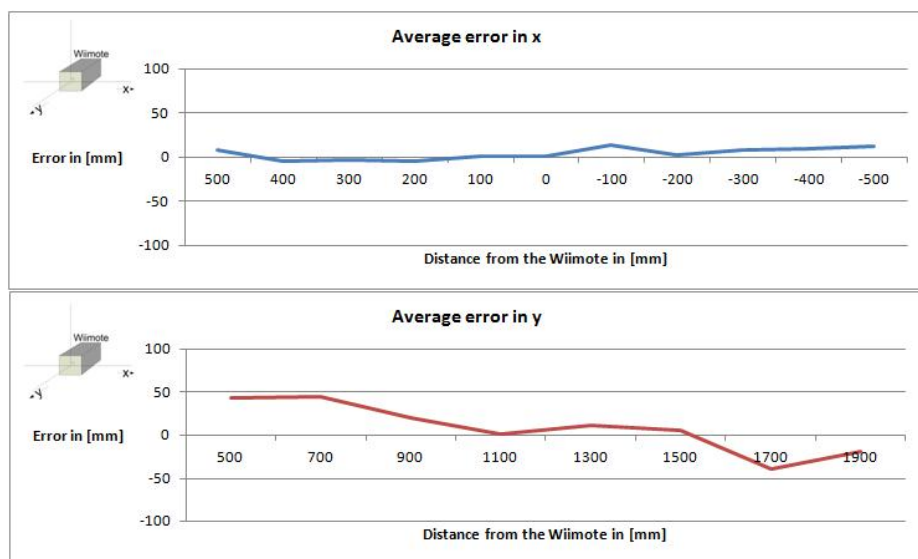


Fig. 2. Wiimote's measurement data ($y=1200$ [mm], height= 420 [mm]) (top), Wiimote's measurement data ($x=0$ [mm], height= 420 [mm]) (bottom)

solution would be to hang the Wiimote from the ceiling. However, this will limit the size of the playing field because of limited field of view of the Wiimote.

3 Framework

The framework needs to accommodate various robots, different sensors and actuators and facilitate communication and cooperation between these devices. Not to be restricted to only one or a set of games/scenarios, the framework should provide basic functionality for defining a game with dynamic rules and strategies. Basically, the framework should be:

- Extendable, to be easy to add new robots, sensor, actuator, scenario, etc.
- Flexible, to support creation of different type of games and support dynamic change of roles, game scenarios, and strategies.
- Platform independent.
- Modular, to be able to just use a sub set of sub-systems and certain aspects to make the framework useable for other games and ubiquitous computing applications.

Since not all robots allow one to directly program them, designing the framework in a completely decentralized approach is rather restricting. On the other hand, designing the framework in a completely centralized fashion is limiting its flexibility and extensibility. Therefore, we opt for a combination of centralized approach (for team strategy and game rules) and decentralized approach (for robot control). In order for this combination to work, in addition to “game” and “robot” layers, an extra layer called Role layer is required. Every team strategy defines a list of roles. Each of these roles defines behavior of a certain robot (e.g. defender, attacker). Every robot can be assigned an initial role by the team strategy. It then executes the commands defined in the role and reacts to sensor input as specified in the role. The robot behavior is therefore decentralized. A robot either changes its role by itself based on sensor input or it

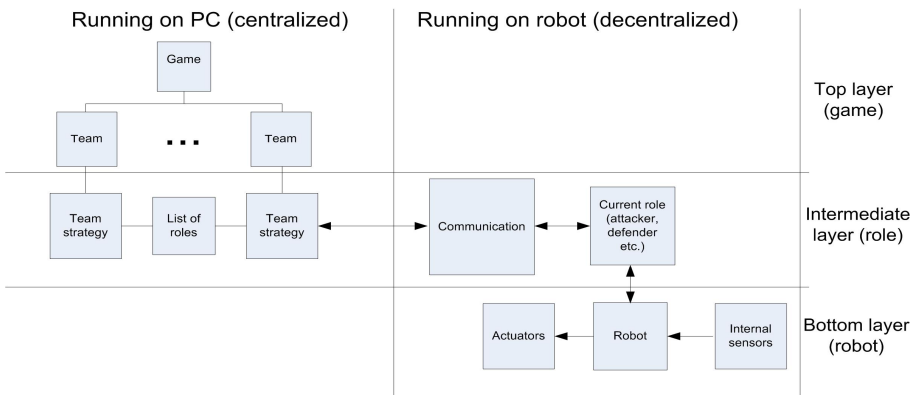


Fig. 3. Diagram presenting different layers of the framework

is given a new role by the team strategy. To ensure the modularity, we use the object-oriented approach to design the framework, which will later be implemented in Java to also assure platform independency. Fig 3 illustrates different layers of the framework.

Robot and Sensor classes assure framework's extensibility, flexibility, dynamicity, and ease of adding new robots, sensors and actuators. These classes contain basic functionality to be used by all types of robots and sensors. This functionality includes communication and identification methods. Robots and sensors are easily implemented by extending these basic classes. The communication protocols for different robots can be placed in separate classes linked to Robot class. This way it will be very easy to use a different form of communication without having to alter the framework. The framework makes a distinction between internal (built-in and embedded in robots) and external sensors (added to the robots, game field, and teams). Internal sensors are integrated in the classes representing the robot, while external sensors are classes extending the basic Sensor class. The reason for doing so is that internal sensors are often read-out by the same protocol that is used to control the robot and therefore it is logical to include them in the Robot class that also governs the protocol for communicating with a particular robot. Moreover, a sensor added to a game serves as a global sensor, which can be accessed by all teams and robots. In the same way a sensor can be added to a team, so that it can be used by all robots in that particular team, or to a robot so that it can only be accessed by that robot. The framework also allows robots to be added to a game instead of a team, so that the robot can function for example as a referee. Fig 4 illustrates the basic class diagram. One should note that actual implementations of game rules, team strategies and roles are placed in classes extending the basic Game, Team and Role classes.

As previously mentioned, using Wiimote requires some form of IR tracking. The framework accommodates this using an object detector, which creates a new instance of a WiiObject class for every source of IR light. As long as the object stays within the field of view (FOV) of the camera, this detector ensures that the WiiObject will always point to the IR source it was created for. The WiiObject instance maintains information concerning the corresponding IR source, such as its position history, number of LEDs an object is represented with, and its last known orientation. Having

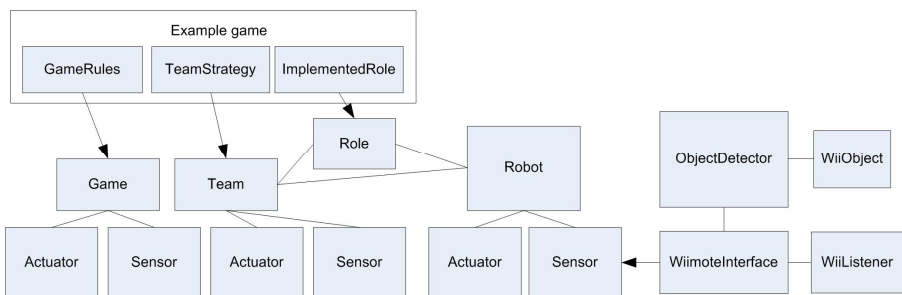


Fig. 4. Simple class diagram

The table below presents the main classes of the framework:

Game	Class that contains list of all teams, robots, sensors and actuators. A class that extends Game contains the rules of a particular game and has the responsibility of creating all sensors and robots.
Team	Class that maintains a list of robots, sensors and actuators belong to a team. Classes that extend Team contain team strategies for a particular game.
Role	Class that represents roles. Every Role is mapped to one robot. Classes that extend Role contain a role designed for a particular robot.
RoleListener	Interface that can be added to an implementation of Team so that it receives events created by roles.
TeamListener	Interface that can be added to an implementation of Game so that it receives events created by teams.
Robot	Abstract class that contains common robot functionality.
Sensor	Class used for external sensors and implements SensorInterface.
Position	Class that represents position and orientation on the playing field.
WiimoteInterface	Class that functions as a gateway between Wiimote and the rest of the framework. It also contains localization techniques. For every Wiimote one WiimoteInterface is needed. It also extends Sensor class.
WiiListener	Class that handles communication with the WiiUseJ library.
ObjectDetector	Class which handles the mapping from raw infrared data to objects on the playing field and creates an instance of WiiObject for every new IR source. It then tracks this source until it leaves the FOV of the Wiimote. It also implements the identification system.
WiiObject	Class which represents an object tracked by ObjectDetector. Maintains information concerning the IR source such as current position, position history and type of object.
EventObject	Every event thrown by a robot or sensor is encapsulated in an implementation of the abstract class EventObject. There are two major types of event objects, TeamEvents and RoleEvent. TeamEvents are handled by TeamListeners (in most cases the implementation of Game), while RoleEvents are handled by RoleListeners (in most cases the implementations of Team).
CommandObject	CommandObject is a class representing a command sent to one of the robots or one of the teams. Examples are StartCommand and GoTo-StartPositionCommand.

all this information in one place makes sharing information among other entities such as teams or robots easy.

When the used identification mechanism does not provide unique identification properties, the following mechanism can be used. When the tracked object goes out of the Wiimotes' FOV, the WiiObject is noted as not being visible. Subsequently when a new object comes within the FOV, this new object is checked against all objects which have previously been visible in order to find a mapping between objects. Parameters such as last seen position and displacement vector are taken into account for this mapping. This way it is possible to track an object while it is being obscured, for example by another robot or object, without creating new WiiObjects every time a robot re-appears or disappears.

4 Game Development

First step to develop a game using the framework is the choice of robots. Figure below illustrates the two robots we have decided to use: Nabaztag (left) and iRobot Roomba (right).



Nabaztag is a simple non-movable gadget whose basic functionality includes reading emails loud, making comments about the weather and talking with other Nabaztags across the Internet. It has rotatable ears (which can only be positioned accurately in steps of 20 degrees in a range of 0° to 180°), several multicolored LEDs and a speaker used for text-to-speech functionality. It also has two sensors, i.e., microphone and RFID reader. Due to limited sensor/actuator capability of Nabaztag, external sensors/actuators need to be used.

Roomba is a circular formed flat vacuum cleaner robot. It contains the following actuators: two motors enabling differential drive, vacuum cleaning brush, speaker, and multiple-color LED. Roomba has 19 sensors, among others, bump detector, touch sensor, dust detector, infrared sensor, angle, and distance sensors.

4.1 Example Game Scenario

Let us consider the following example game scenario. There are two teams, an attacking team and a defending team. Each team consists of a Nabaztag and a Roomba. Each Nabaztag has a Wiimote mounted on top. The goal for the attacking team is to touch the defending Nabaztag with its Roomba. The goal for the defending team is to prevent the attacking team from succeeding. Defending can be done by touching the attacking Roomba. Both teams have their own Wiimotes, which they use for locating their own as well as the enemy Roomba. When attacking Roomba touches defending Nabaztag or defending Roomba touches attacking Roomba, teams switch and the game starts all over again.

Due to the fact that the presented framework provides a solid base and modules, implementing specific games require very limited additions. For instance, for our example game scenario, we only need to create an extension to the Game class to accommodate the rules for this specific game and to create two different team strategies, each containing several roles and set up the two different robots.

4.2 Technical Problems and Solutions

To attach the Wiimote to the Nabaztag, a Lego construction is attached in place of Nabaztag's ears. The ear controller controls a gearbox (to increase the accuracy of the

rotatable ears) and a turntable running on bearings on top of the construct. This construction results in a Nabaztag which is able to turn its ‘head’ (and also the Wiimote) accurately with a 22.5° resolution.

To control the Nabaztag, a communication class called NabaztagComm is created that, among other things, sets the position of the ear between 0° and 180°, sets LEDs with any given color, sends the Nabaztag to sleep and wakes it up. Moreover, the WiiNabaztag class is created, which represents a special case of the robot that is equipped with the Wiimote. This class rotates the ears and keeps track of orientation of the Wiimote attached to it. It also supports setting the Wiimote direction in 16 different directions between 0° and 360°.

Although Roomba’s angle and distance sensors are useful in determining robot’s position and orientation, their values seem to be incorrect. After conducting various experiments, it became apparent that there is neither constant factor between the actual and theoretical number nor is the error systematic. Another problem of Roomba is that it is not possible to program it directly as it has no internal memory that can be accessed.

When the Roomba is supposed to drive a certain distance, the library sends a drive command to the Roomba. At this moment a timer is started. After a given time, which is determined by the specified distance and speed, the library sends a stop command. This mechanism is not very accurate because of the exact timing involved. Using the library in a heavily multi-threaded environment, this becomes a serious problem. In order to determine the magnitude of this problem a couple of tests have been performed.

Table 1 presents results of various tests in which the Roomba was driven certain distances. A number of observations can be made from these experiments. First, the average errors compared to the driven distances are very small. Secondly, the relatively large gap between the average deviation and the maximum deviation shows that timing is indeed important. Thirdly, Roomba necessitates good calibration. Experiments show that speed 110 [mm/s], which is not as well calibrated as the other two speeds, results in significantly larger errors.

Table 1. Roomba distance measurements (left), Roomba Angle measurements (right)

Speed [mm/s]	Desired distance [mm]	Average deviation [mm]	Max. deviation [mm]	Average error/meter [mm/m]
110	1000	9.3	12	9.3
	2000	18.3	29	9.2
	3000	27.0	41	9.0
194	1000	8	16	8
	2000	10.6	20	5.3
	3000	13.3	20	6.7
305	1000	5.7	7	5.7
	2000	5.0	10	2.5
	3000	2.0	3	0.7

Desired rotation [°]	Average deviation [°]
360	1.4
-360	0.4
720	1.6
-720	1.6

Because a straight line is not the only path a Roomba is required to traverse, some extra tests were performed in which the Roomba was required to turn specific angles. The tests were performed in both clock-wise and anti clock-wise directions to see if there was difference between the motors driving different wheels. As the results presented in Table 1 show, the error made when rotating the robot is very small. Control of the Roomba's movement after the calibration and under aforementioned circumstances is quite precise.

To implement the example game scenario, we use a simple identification technique in which each robot of one of the teams was equipped with a single LED and each robot of the other team with two LEDs. In addition to this, the framework used input from robot's angle and distance sensors when they were out of view of the Wiimote.

5 Conclusion

The generic framework presented in this paper is able to accommodate various robots, sensors and actuators. Its flexibility, extensibility and modularity enable developing a broad range of pervasive games and game scenarios based upon robots, sensors, and actuators. To demonstrate the framework an example game using two commonly available robots has been developed. Furthermore, the cheap and easily accessible Wiimote-based localization and identification technique presented can be used in variety of ubiquitous systems including pervasive games. The accuracy of the proposed technique is good for distances up to three meters and is comparable with existing infrared beacon triangulation methods. At longer distances, however, accuracy decreases.

To enable controlling the robots more accurately, the future work includes integrating the framework with a full feedback controller using the localization system as input. Another possible extension to the framework is support for more complicated movement patterns such as spline, which will allow more advanced strategies.

References

1. Weiser, M.: The Computer for the Twenty-First Century. *Scientific American*, 94–110 (1991)
2. Schneider, J., Kortuem, G.: How to Host a Pervasive Game: Supporting Face-to-Face Interactions in Live-Action Roleplaying. In: *Proc. Designing Ubiquitous Computing Games* (2001)
3. Magerkuth, C., Stenzel, R., Streitz, N., Neuhold, E.: A Multimodal Interaction Framework from Pervasive Game Applications. In: *Workshop for Artificial Intelligence in Mobile Systems, USA* (2003)
4. Anastasi, R., Tandavanitj, N., Flintham, M., Crabtree, A., Adams, M., Row-Farr, J., Iddon, J., Benford, S., Hemmings, T., Izadi, S., Taylor, I.: Can You See Me Now? A Citywide Mixed-Reality Gaming Experience, *Equator Technical Report*, University of Nottingham (2002)
5. Björk, S., Falk, J., Hansson, R., Ljungstrand, P.: Pirates! –Using the Physical World as a Game Board. In: *Proc. Interact 2001, IFIP TC.13 Conference on Human-Computer Interaction, Tokyo, Japan* (2001)

6. Linner, D., Kirsch, F., Radosch, I., Steglich, S.: Context-aware Multimedia Provisioning for Pervasive Games. In: Proc. of the 7th IEEE International Symposium on Multimedia (2005)
7. Benford, S., Magerkuth, C., Ljungstrand, P.: Bridging the physical and digital in pervasive gaming. *Communications of the ACM*, 54–75 (March 2005)
8. Brassart, E., Pegard, C., Mouadibb, M.: Localization using infrared beacons. *Robotica* 18 (2000)
9. Mottaghi, R., Vaughan, R.: An integrated particle filter and potential field method applied to cooperative multi-robot target tracking. *Autonomous Robots*, 19–35 (July 2007)
10. Eom, D., Jang, J., Kim, T., Han, J.: A VR Game Platform Built Upon Wireless Sensor Network. In: *Advances in Visual Computing*. Springer, Berlin (2006)
11. Weiss, N., Hildebrand, L.: An Exemplary Robot Soccer Vision System. In: *Workshop on Robots in Entertainment, Leisure and Hobby, Austria* (2004)
12. http://en.wikipedia.org/wiki/Wii_Remote
13. Laforest, M.: Wiiuse, <http://www.wiiuse.net>

HTTPStream Platform – Low Latency Data for the Web

Marios Tziakouris and Paraskevas Evripidou

University of Cyprus, Computer Science Department, Nicosia, Cyprus
Marios.Tziakouris@cse.com.cy, skevos@cs.ucy.ac.cy

Abstract. Timely delivery of information and live updates are of paramount importance in our connected society. Despite its tremendous penetration and wide acceptance, the Web failed to move to the next level and provide the means for changing the way sophisticated applications are delivered to the users. One of the major issues is its inability to provide low-latency (real-time) data and notifications to web applications with frequent data changes. In this paper we present the HTTPStream platform which aims in delivering low-latency data to web applications utilizing generally accepted web principles. The platform is based on the concept of establishing a permanent connection between the server and the client by “trapping” the server response into a non-ended loop and utilizing it to stream fresh data to the browser. The HTTPStream server complements conventional web servers by handling only the low latency data.

Keywords: Web 2.0, low latency data, http streaming, pervasive.

1 Introduction

The promise of the Web as the easily distributed and managed, ubiquitous application User Interface (UI) never came to absolute fruition. While it worked well for brochure-ware and simplistic transactional systems, it was not able to meet the challenge of the first generation of rich clients in delivering rich, flexible, high-performing and easy-to-use desktop applications [1]. Although the simplicity of the Web’s core technologies (HTML, HTTP and URI) allowed for its wide adoption, at the same time it significantly hindered its penetration to the sophisticated distributed applications platform area. Even though having a common language for describing the user interface is a great concept, it appears that the limitations of HTML hinder significantly the Web applications in providing functionality and usability to the levels of their desktop counterparts [23]. Similarly the pull-only orientation of the HTTP communication protocol does not allow the developers to provide timely, uninterrupted data updates and notifications to the users.

Efforts for a richer Web application experience are currently concentrated on better exploiting the existing technologies. The stateless nature of the web, the pull-only orientation of the request-response model and the need to reload the whole page for almost every user action, gave rise to efforts related in improving the implications of having numerous round-trips to the server for accomplishing simple tasks. Some of these efforts are focusing on optimizing the data retrieval process by leveraging the

browser's JavaScript engine to render user interfaces without reloading pages (single-page level programming). Asynchronous JavaScript + XML (AJAX) technology is the latest approach in providing enhanced usability and interactivity to web applications. The intent of AJAX is to make web pages feel more responsive by exchanging small amounts of data with the server in background, so that the entire web page does not have to be reloaded each time the user requests a change.

AJAX does not address fully the needs of web applications like stock trading systems, online social networks, news portals and primarily pervasive applications. Essentially, such applications with frequent data updates need to have the server initiating the data transfer to the client, reversing effectively the web paradigm which calls for a request-response model (client to server). Developers have experimented with various approaches and have employed several technologies trying to provide solutions for providing "push data" functionality to the web (or an emulation of "push data"). It appears that only solutions that adhere to generally accepted Web principles can be sustainable and be adopted by the web community.

Pervasive services have to be interoperable with web technologies in order to fully deliver their services anytime, anywhere and on the most suitable devices at each context. Mobile devices today are getting more powerful and can handle more complex computing tasks. Many of issues such as power consumption, speed, storage and connectivity have been overcome. Many established web services are aiming to offer their services on the go.

Motivated by the lack of a comprehensive study on this area, our work aims in proposing an alternative approach for providing real-time data over the Web and an associated platform for leveraging this approach. The structure of this work is as follows: In the next section we discuss about providing real-time data to the web using the server push approach and the difficulties to implement this approach over the web. In section 3 we present our proposed platform based on our concept and we elaborate on its advantages and disadvantages. Section 4 describes an initial prototype of our platform and refers to relevant tests that prove its effectiveness in providing real-time, asynchronous data to web applications. Finally we conclude in Section 5 including our thoughts for future work.

2 Real-Time Data

Although Web 2.0 technologies have paved the way for a richer web application experience, there are some types of applications that are not benefited as much as necessary in order to match their desktop counterparts or in order to fulfil the needs of pervasive applications. Applications such as stock trading, social networking and primarily pervasive applications that require real-time data updates and asynchronous messaging are having a hard time in unlocking their full potential if deployed using web technologies. For today's users single interaction updates are not enough and users in the same "context" need live updates of the changes others make instantly. Moreover, there is the need of supporting user's mobility. A user working with a desktop-based stock monitoring application needs to be able to continue receiving stock information even when he is on the move.

Researchers and developers have experimented with various approaches trying to minimize the latency of the data updates using web technologies. The most predominant approach is the push model. Implementing push technology over the web is not a straightforward task given the pull-model orientation of the web and the absence of any standards to this direction. Effectively the push technology over the web calls for a reverse of the web paradigm where the client becomes a passive part of the system and receives the updates from the server as soon as they are available.

The push model is the ideal method for delivering real-time data to web applications. It is of course the proposed solution to any real-time traditional application but the fact that it is has to run over the web make its implementation rather difficult. With this approach the data delivery is completely asynchronous with respect to the user and the browser. This results in virtually a real-time system with the lowest latency possible and on top of that with the smaller disruption of the web infrastructure. It is implemented by establishing a permanent connection between the server and browser which the server utilizes to send updates and notifications to the user as soon as they happen.

The selection of server push as the favourable approach for delivering real-time data over the web must not come with surprise. The problem is that the core concepts of server push do not really apply over the web. The web is considered an unreliable network and data delivery is not guaranteed (best effort delivery). Not only this, the web's transport protocol is not designed to support these interactions and also there is lack of control of the client side i.e. the application runs in the browser and the developers do not have full control of what a browser can do. Although there were a lot of efforts to deliver real-time data to web applications and most of them have been technologically sound, none of them have really managed to be widely adopted. The main reason is that all of them failed to adhere to the web standards. Most of them require special plug-ins to be installed on the client [7], others are requiring extensive changes to the web network [12] [13] and others assume a complete overhaul of the complete web ecosystem [14]. It is generally accepted now that with the wide acceptance of the web and the vast installation of web applications around the world, it is very difficult to alter significantly the core elements of the web. Thus, any new approach to be adopted by web developers has to be aligned with the following principles:

- Implement an HTML interface: The output of the application must always be able to be rendered by a browser. This excludes plug-ins interfaces such as Macromedia's Flash [7]
- Immediate Start: The web application must start as soon as the user hits the URL.
- Traverse firewalls and proxy systems: This implies adherence to the commonalities of the HTTP protocol especially with respect to the ports used (80,443).
- Cross-browser: The web application must be rendered well in all popular browsers and it has to implement functionality that is present to all browsers
- Acceptable security risks
- No plug-ins: The usage of plug-ins is not favourable by most users

Our proposed HTTPStream platform is completely aligned with the aforementioned principles and aims in delivering low-latency data over the web with no compromises.

3 The HTTPStream Platform

The HTTPStream platform consists of 3 core elements as shown in figure 2: the cross-browser client engine, the “HTTPStreamer” web server and the data watcher service.

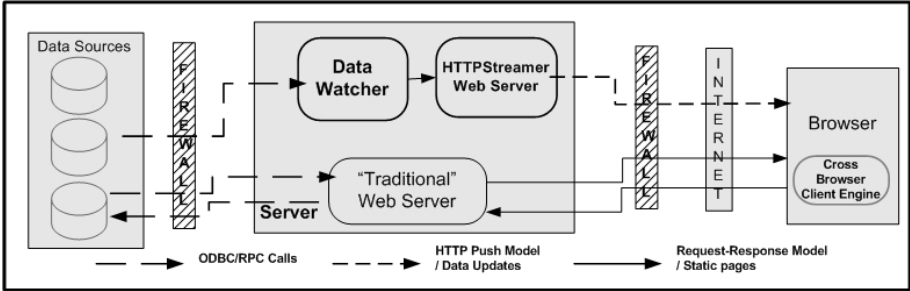


Fig. 1. The HTTPStream platform

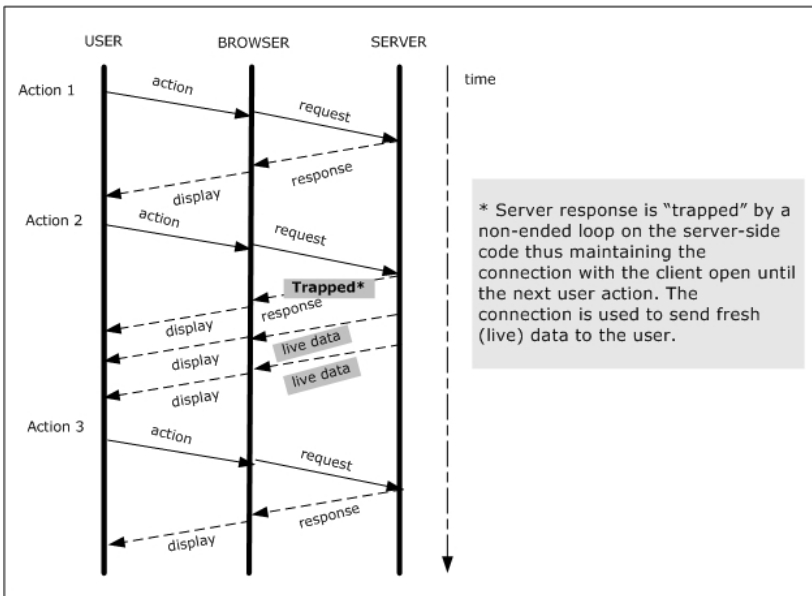


Fig. 2. Core concept of permanent connection of HTTPStream platform

The platform is based on the concept of implementing a permanent connection between the server the client (figure 2). The platform is designed so that it can co-exist with current web servers and will kick in only when information with the need of real-time data is requested. All other queries will be served as usual by the “traditional” web server and will not make any use of the cross-browser client engine. The notion

of permanent connection is not a straightforward task in the web environment if one has to adopt accepted web principles. Our approach, which is central to this platform, is to implement this connection by maintaining a long-lived connection between the server and the client by “trapping” the server response into a non-ended loop, keeping in that way the channel open “indefinitely” (figure 3). In other words, we are preventing the server to execute the connection-header “close” that is always send by the browsers to signal the end of the persistent connection between the client and the server, thus preserving the connection between the two open continuously.

Although this seems an unusual approach, the truth is that it can be done and on top of this it is completely aligned with the aforementioned web principles. Basically this consists of implementing a non-ended loop on the server-side and within that loop fresh data is streamed to the browser. Note that the always open connection stems from an AJAX call that gives more flexibility in terms of the length and frequency of connections than using the regular HTTP GET request. It allows for refreshing the connection once in a while and it carries only the necessary data payload. Of course there are downsides to this approach but the thing is that it works, it capitalizes on the current web technologies and its problems can be mediated. The HTTPStream platform is based on this approach and aims in providing solutions to these problems.

On the client side, the HTTPStream platform employs a cross-browser client engine which is responsible for implementing the concept of permanent connection in co-operation with the “HTTPStreamer” web server. Although the concept calls for the server to maintain the connection, the client side is also important. First of all it has to be cross-browser for compatibility and portability reasons. It is build around Web 2.0 concepts so that it maintains the nature of page level programming and minimizes unnecessary traffic to the minimum. The client side is responsible to inform the server about the status of the connection and in cases the connection is dropped (user action) the client will request a re-connection (recycle). Additionally the client engine handles the de-serialization of the updated data structures (XML or Javascript primitives) and applies the necessary changes to the page. The whole process is totally asynchronous in all its aspects. Moreover the client engine is responsible to inform the server for various “accounting” information such as bandwidth information for server streaming adaptation, network status (congestion) and user activity.

The server part is the most complex part of the platform. Not only it has to perform its regular functions (connections handling, business logic processing, database access, security), it also has to implement the “permanent” connection concept and the need to “stream” data to the client. For that reason we have decided to separate these functions and regular web server functions are served by a traditional web server and all “streaming” functions are handled by an “HTTPStreamer” web server optimized for the needs of providing real-time data.

Apparently web servers are quite stressed under the concept of server push. Permanent connections are not ideal for web servers given that traditional web servers consume fixed resources per request. The server limit is reached quickly and web servers cannot cope with the increased load. For that reason we propose the use of an “HTTPStreamer” web server customized for the needs of this concept. This server must have the following characteristics in order to address the scalability issue:

- Allow direct control of the TCP/IP stack to handle various tasks such as connection handling, streaming, bandwidth allocation e.t.c
- Based on an event driven process (daemon) that will generate and deliver data to multiple clients based on events (update events) without allocating a thread for each connection
- Decouple of the number of connections from the number of threads available
- Small memory footprint
- Effective management of CPU
- Support for asynchronous I/O
- Support for clustering

We expect that a server build with these characteristics in mind will be able to serve thousands of “permanent” connections and will address the scalability issue sufficiently. The idea is to build a server that internally handles the preparation of “interesting” data (channels) once and push to many clients within a single, light-weight and dedicated server process. Similar concepts are found in several research and commercial efforts such as kqueue (FreeBSD) [3], epoll (Linux) [6], POE (Perl) [10], Twisted (Python) [11], event_mpm (Apache 2.2) [2] and Jetty (Java) [5]. None of these efforts are addressing the issue exactly the way we are approaching it but they include concepts and technologies that can contribute to the implementation of the “HTTPStreamer” web server. This is part of our future work.

In addition to the problems arising from the scalability concerns the “HTTPStreamer” web server must provide functionality regarding bandwidth control and adaptive streaming. By utilizing “accounting” information from the client the server has to allocate bandwidth according to availability and adapt the data streaming according to bandwidth availability / client capability. Heuristics mechanisms will be employed in order to adapt the data streaming without compromising the data coherency and without saturating the network. This functionality is crucial in such environments since any unnecessary, not optimized process might result in total system failure.

The last component of the HTTPStream platform is the Data Watcher service. This service is deemed as necessary because we want to be able to control the data flow from the source to the final destination. This service complements the asynchronous approach of the rest of the platform. It would not be sensible to employ all these techniques to stream live data from the web server to the client and have stale data within an environment that is completely controllable by us. Therefore, the Data Watcher service aims in further optimizing in terms of latency the data transfer to the client and in addition it is used in order to maintain the clients’ registrations to the available data channels.

4 Platform Prototype Testing

In order to prove the effectiveness of our approach in terms of providing real-time data we tested a prototype of our platform against currently used methods. Specifically the prototype was tested against the classic page refresh method (full page reload) and the more recent periodic polling method which is based on an AJAX call. The prototype used includes only the necessary components to implement the concept

of permanent connection. On the client-side the prototype include the necessary functions for managing the AJAX connections to the server (creation and recycling as necessary) as well as the receiving, de-serialization and parsing of the results. It also provides adaptation for major browsers and basic error handling regarding connectivity loss. A part of the client side pseudo-code is shown below.

```
function checkForNewdata(){
createAJAXRequest() // Create an AJAX request
if (rqst) {
rqst.onreadystatechange=processReqChange;//Request status
rqst.open("GET", url, true);
rqst.send("");}

function createAJAXRequest() {
// branch for browsers (native)
if(window.XMLHttpRequest && !(window.ActiveXObject))
{rqst = new XMLHttpRequest();}
else if(window.ActiveXObject)
{rqst = new ActiveXObject("Msxml2.XMLHTTP");}
else
{rqst = new ActiveXObject("Microsoft.XMLHTTP");}

function processReqChange()
{if (rqst.readyState == 3) // if something is received
{ProcessInput(rqst.responseText);
// At some (arbitrary) length recycle the connection
if (rqst.responseText.length > 3000)
{checkForNewdata();}

function ProcessInput(input)
{ //do something with the response
var out = document.getElementById('outputZone');
out.innerHTML = lastMessage;}
```

On the server side, the tests were run on a classic Microsoft Internet Information Server 7.0 (IIS) and the server-side development framework used was ASP.NET 2.0. No optimizations or adaptations for the implementation of the permanent connection were made on the server. A part of the server pseudo-code is shown below.

```
// connect to the datasource (can be anything)
conn.Open()
While (True) // Implement non-ended loop
    Response.Clear() //clear buffers from previous request
    // check for new data
    SQLstr = "SELECT * from DataSource"
    NewRecords = GetNewRecords(SQLstr)
    If NewRecords Then
```

```

While NewRecords
    output = output & NewRecords
    Serialize(output) // i.e. XML, JSON, text
End While
Response.Write(output) // write output to buffers
Response.Flush() // flush buffers
End if
End While
conn.Close()

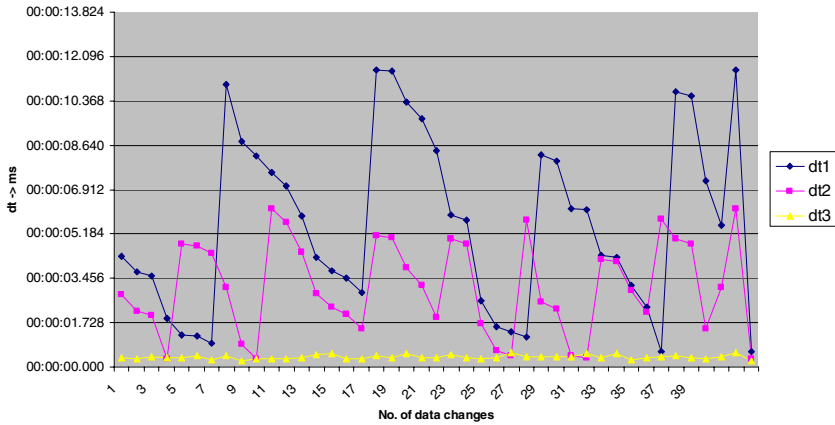
```

The data watcher service was not used at all. The prototype was reading directly from a data-source that was constantly changing through a separate process.

The three approaches were tested on the same equipment. The code for the classic refresh and the periodic polling method is written using the same development platform to ensure the homogeneity of the tests. In order to measure the latency between the data generation and the data arrival to the user we tracked the point in time where the data is generated (through a separate process) as well as the point of time the data was displayed on the user's browser. The results are plotted in the following graph. The X-axis illustrates the number of data changes and the Y-axis the difference between the generation and display of the data on the browser. For the classic refresh method a refresh time of 10 seconds has been selected which is the absolute minimum that can be used. Usually the refresh intervals of the classic method are between 30 seconds to 5 minutes due to the need to reload the whole page. In the case of periodic polling, since the server call is an AJAX call, which is substantially more lightweight from the classic refresh, a 5 second interval was used. In the HTTPStream platform the concept of refresh interval simply does not exist, since the server is responsible for sending the data to the client.

Obviously the results are as expected. The server push concept implemented by the HTTPStream platform far outperforms the other two methods. This is evident not only from the graph, that plots the latencies between the three methods, but also from the statistics table where the average latency of the classic refresh method was about 5.5s, for the periodic polling method about 3.1s and for the server push method only 0.4s. On top of that, the maximum latency recorded for the classic refresh method is 11.5s which is even more than the refresh interval. This is also true in the case of the periodic polling where the maximum latency was 6.1s. This discrepancy stems from the fact the except the polling interval, in the first two methods the TCP and HTTP request setup is added in the latency as well as the browser rendering which in the case of the classic refresh is significant since the browser has to re-render the whole page again. In the case of HTTPStream the connection is permanent and thus no connection establishment takes place.

Finally, in order to illustrate the adverse effect of the HTTPStream approach on the web server we recorded values of the CPU performance of the server. Using only one client each time, in the classic refresh approach the server was using a 5% of the processing power in intervals of approximately 10 seconds. In the case of periodic calling the server was utilizing a 2% of the processing power in intervals of 5 seconds and in the case of HTTPStream prototype the server was constantly using a 3% of the



dt1: Refresh Method (10s) dt2: Periodic Polling (5s) dt3: HTTPStream

Fig. 3. Latency Times

Table 1. Latency Statistics

	dt1	dt2	dt3
Average	00:00:05.533	00:00:03.105	00:00:00.388
Max	00:00:11.568	00:00:06.184	00:00:00.566
Min	00:00:00.623	00:00:00.326	00:00:00.231

CPU. Apparently, the server resources limit will be reached much earlier in the case of the HTTPStream platform than in the other cases. This underlines the need for the development of a custom web server for the needs of this approach.

5 Conclusions

The HTTPStream platform aims to provide low-latency data to web and pervasive applications utilizing the web as an application platform. It is completely aligned with documented and undocumented web principles and imposes only minimum disruptions to the current web infrastructure. Additionally, it allows for the asynchronous nature of pervasive applications as well as it provides a sound notification mechanism necessary for such applications. It is our belief that the platform can bring closer the worlds of pervasive and web computing and will allow for the easy entrance of popular web services in the pervasive world.

The HTTPStream platform is adopting a server push approach to push data to the client. The key concept is the establishment of a permanent connection between the server and the client using web accepted principles. The platform aspires to provide

solutions to the problems of the solution the most notable of which is the need for an “HTTPStreamer” web server that will address the scalability issue.

Currently only the cross-browser client engine is significantly developed. Some parts of the “HTTPStreamer” web server are also ready. Our future work aims in completing the work for the “HTTPStreamer” web server with all of its components along with the necessary testing. Additionally, we plan to showcase applications that span the web and pervasive world. The development of the Data Watcher will be done last since the service does not constitute a major research task but is needed to complement the asynchronous nature of the platform.

References

1. Lopez-Ortiz, A., German, D.: A Multicollaborative Push-Caching HTTP Protocol for the WWW (2001)
2. Apache Foundation, event_mpm, http://docx.itsscales.com/experimental_2event_2mpm_8h-source.html
3. FreeBSD.org, Kqueue: A generic and scalable event notification facility, <http://people.freebsd.org/~jlemon/papers/kqueue.pdf>
4. Garrett, J.J.: Ajax: A New Approach to Web Applications, <http://www.adaptivepath.com/publications/essays/archives/000385.php>
5. Jetty, <http://www.mortbay.org/>
6. epoll: I/O event notification facility, <http://linux.die.net/man/4/epoll>
7. Macromedia Flash, Create rich content and applications across desktops and devices, <http://www.macromedia.com/software/flash/>
8. Rees, M.J.: Evolving the Browser Towards a Standard User Interface Architecture. In: 3rd Australasian User Interface Conference, pp. 1–8 (2002)
9. Strahl, R.: Diminishing Importance of HTML, <http://www.west-wind.com/presentations/Editorials/DiminishingImportanceOfHTML.asp>
10. Perl.org, POE: Perl Object Environment, <http://poe.perl.org/>
11. Twisted Matrix Labs, Twisted: an event-driven networking engine written in Python, <http://twistedmatrix.com/trac/>
12. Ammar, M., Almeroth, K., Clark, R., Fei, Z.: Multicast delivery of web pages or how to make web servers pushy. In: Workshop on Internet Server Performance, Madison, Wisconsin (1998)
13. Almeroth, K.C., Ammar, M.H., Zongming, F.: Scalable delivery of web pages using cyclic best-effort (UDP) multicast. In: INFOCOM 1998. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies (1998)
14. Deolasee, P., Katkar, A., Panchbudhe, A., Ramamritham, K., Shenoy, P.: Adaptive Push-Pull: Disseminating Dynamic Web Data (2004)

Password Streaming for RFID Privacy

Victor K.Y. Wu and Roy H. Campbell

University of Illinois at Urbana-Champaign, IL, USA
{vwu3,rhc}@illinois.edu

Abstract. We propose a novel privacy-protecting RFID system using *password streaming*. Our scheme assumes a centralized back-end system containing a database of tag passwords. A reader uses this database to broadcast password “guesses” continuously in its domain, without it needing to scan for the presence of tags. A tag that receives a matching password tells the reader it is present, and the password is replaced with a new one that is synchronized both in the tag and the database. Our scheme also assumes multiple readers with their respective domains in a large physical space. As tags and attackers (adversaries intent on compromising privacy) move between domains, the back-end system coordinates the readers’ password streams to defend against attacks. Our scheme pushes the computational complexity to the back-end system, very much in the spirit of RFID. It defends against stronger types of attacks than those in [4].

1 Introduction

In an RFID (radio frequency identification) system, RFID tags carrying unique tag IDs (identifiers) are affixed to objects, usually one per object. An RFID reader queries a tag by transmitting a wireless signal, to which the tag responds with its tag ID. The reader then uses the tag ID as a pointer to access a database to quickly determine potentially a plethora of information regarding the object. Unfortunately, the presence of malicious entities (which we call attackers) precludes the *privacy* of this object information. In this paper, we focus on defending against *inventorying* and *tracking* of tags to protect privacy [1]. Inventorying refers to an attacker acquiring a tag ID, and using it to learn information regarding its associated object. Tracking refers to an attacker physically (or otherwise) following a tag by successively querying it, also to learn information regarding the associated object. Tracking is still possible if a tag responds with a meta identifier instead of its tag ID.

In this paper, we seek a solution to the privacy problem. Since RFID tags (especially passive tags that we exclusively consider) are severely constrained in their computational power, our solution must push most of the computational requirements to the readers and their supporting infrastructure. A seemingly plausible solution is the use of password-specific tag IDs. That is, to access a tag, a reader has to first authenticate itself by providing the tag’s password. Presumably, the password associated with this tag’s ID is initially stored in a

database that the reader has access. But to know which password to use, the reader has to first know the tag's ID, which if it knew, would defeat the goal of the solution. In other words, we have a chicken-and-egg dilemma [2]. We mention current solutions in the literature that steer clear of this problem. We then offer a novel solution and discuss its advantages.

The authors in [2] use a password-less system based on *secret sharing* [3]. In a (t, n) sharing scheme, a secret is divided into n shares, which are distributed. If at least $t \leq n$ shares are recovered, the secret can be reconstructed. Anything less than t shares reveals nothing about the secret. A tag ID is first divided into its secret shares. The shares are concatenated, and the resulting string replaces the tag ID in the tag. A reader querying the tag receives random bits from the string with the position of the bit indicated. Eventually, enough shares are collected, and the secret tag ID can be reconstructed. The process is accelerated by using caches in the reader. An attacker is allowed only a limited number of successive queries due to throttling, an assumption of the weakened attack model of [4]. This scheme is indeed strong. Nonetheless, if we do allow an attacker to query a tag for a long and undisturbed period of time, the attacker does finally acquire the tag ID. The weakened attack model may allow a reader and a tag occasionally to talk to each other without attacker interference, to perhaps update the secret shares. But if we adjust the attacker model slightly to allow for another attacker, for example, to listen to that communication as well, the scheme fails. Therefore, the password-less nature of this scheme is not necessarily effective in stronger attack scenarios.

The authors in [4] use a password-based system. Rotating *pseudonyms* transmitted by a tag are used to confuse attackers, and are mapped by legitimate readers to the desired unique tag ID. Specifically, a tag responds to a reader query with a pseudonym α_i , and the reader maps the pseudonym to a tag ID. The reader then authenticates itself by sending a password β_i , unique to α_i . Then, the tag authenticates itself to the reader with a password γ_i , unique to α_i . Finally, α_i , β_i , and γ_i are updated using an exclusive or (XOR) one-time pad algorithm, that we borrow and describe in Sect. 4.1. The algorithm guarantees that for an attacker to learn anything about α_i , β_i , and γ_i , it must listen to m successive reader-to-tag communications, where m is a system parameter. This scheme suffers from being inefficient. First, it requires many communication flows to establish privacy. Secondly, the pseudonyms impose additional memory constraints on the tags.

In this paper, we propose a novel privacy-protecting RFID system using *password streaming*. We avoid the chicken-and-egg problem by eliminating the initial flow of a reader “knocking on the door” of a tag. Instead, our proposed scheme assumes a reader has a good approximation (based on information provided by a centralized back-end system) of the tags in its domain, and immediately starts streaming the passwords of those tags, in a broadcast fashion, allowing for the tags to respond in between passwords. A tag that receives a matching password updates its local copy by randomly generating a new password to replace the existing one, and sends back the update to the reader. The reader thus confirms which tag

is present in its domain. The back-end system coordinates the password streams of multiple readers in their respective domains. Thus, tag passwords and locations are closely tracked. As a result, a “fresh” password that no tag accepts in a domain is quickly rendered “stale”, since the password is quickly updated in another domain where the tag has moved. This prevents an attacker to use the stale password to compromise privacy. In particular, the main contributions of this paper include the decision criteria for and choice of the passwords that are streamed.

Note that an immediate advantage of our scheme is that it is a singulation algorithm, although very different from the traditional tree-walking [5] and Aloha-based [6] algorithms. That is, we do not require privacy to be a separate communications layer. Our proposed scheme is efficient and light-weight. In Sects. 3 and 4, we point out the low computational requirements of tags, very much in the spirit of RFID. This is because our proposed scheme relies heavily on the supporting infrastructure of the RFID system. RFID technology emphasizes low-cost tags, and that complexity should be pushed to the readers. We believe that this is true also for privacy, and therefore push computational requirements to the centralized back-end system. This structure allows our proposed scheme to defend against stronger types of attacks than those in [4].

The rest of the paper is organized as follows. Section 2 sets up the system model. Section 3 gives a generalized description of our proposed scheme. In Sect. 4, we refine our scheme to defend against increasingly stronger attacks. Section 5 describes how the supporting infrastructure of an RFID system can be explored in relation to privacy. Finally, Sect. 6 concludes the paper, and offers future research directions.

2 System Model

We define the components in our privacy system.

- **tag:** A tag refers to a legitimate passive RFID tag affixed to a physical object by the user (or application) of the privacy system.
- **reader:** A reader refers to a legitimate RFID reader. All readers are authorized to interrogate any tag. All readers are controlled by the centralized back-end system.
- **back-end system:** The back-end system is the abstract upper layer that sits above the physical and data link layers of the tags and readers. It coordinates how readers interact with tags in their respective domains, as well as manage tag IDs and passwords. It thus provides a privacy-protecting RFID service to the user (or application).
- **attacker:** An attacker is a malicious entity that can interact only with tags and readers wirelessly, using radio frequency signals. It may be stronger or weaker than a reader in terms of transmitting, receiving, and processing these signals. It may even physically attack tags or readers. It does not have access to the back-end system. The back-end system also cannot detect attackers directly.

Our system model assumes a large physical space where readers are initially distributed throughout. The readers are used to singulate tags in their respective wireless ranges, which we call domains. The readers remain stationary, and their domains do not overlap. Each reader has a direct connection to the back-end system (via a dedicated wifi access point using a separate channel from RFID communications, for example). Tags are also distributed throughout the physical space. They are mobile, and each tag may at most be in one domain at any time instant. Wireless signals from readers and tags are confined to their current domain, and do not interfere with the wireless signals from other domains. Attackers are mobile, and can simultaneously communicate with readers and tags in at most one domain. We also assume some minimal level of communications and physical security. That is, attackers wirelessly communicating with each other, or otherwise transmitting with large powers, are eventually detected. Attackers in the physical space and even just outside the boundary, are eventually caught if they do not leave after a relatively short period of time. Too many of them, especially if they are in motion, also lead to their quick capture.

3 Proposed Scheme

We describe our proposed scheme, in a purposely generalized notion. In Sect. 4, we provide specific details according to increasingly stronger attacker models and requirements of the application.

Every tag in the physical space has a unique and dynamic password stored in its memory. As well, the passwords are mirrored in a database in the back-end system, each associated with a tag's ID. That is, tag passwords are synchronized between the back-end system and tags. Our proposed scheme is actually a singulation algorithm for the readers described in Sect. 2. In each domain, the reader streams tag passwords (chosen from the back-end system) continuously, in a broadcast fashion. Time is allowed between each streamed password for tags to respond. When a tag receives a password that does not match its own password, it responds with a common (across all tags) and pre-determined dummy signal. When a tag does receive a matching password, it immediately updates its local copy by randomly generating a new password to replace the existing one, and sends back the update to the reader. It then goes into sleep mode (that is, does not respond to any reader queries) until the reader has finished the current singulation session. Therefore, when a streamed password does not match, the reader receives a summation of dummy signals due to multiple responses from multiple tags. The reader receiver is designed to handle multi-path propagation at the physical and data link layers. This allows the reader to interpret the identical dummy signals as multi-path copies of a single dummy signal, and it can thus conclude that the tag associated with the password it had streamed is not present. When a streamed password does match, the reader interprets the responses as the summation of two signals. One is a dummy signal experiencing multi-path propagation while the other is a password update signal. Therefore, the reader can subtract out the dummy signal from the summation to extract the updated password, and pass on the update to the

back-end system to maintain synchronization. The associated tag is thus singulated. A singulation session is complete when all tags have gone to sleep, and the reader no longer receives any responses.

The main contributions of this paper include the decision criteria for and choice of the passwords that are streamed. Since the password space is large, the solution is not immediate. For example, consider the reader first choosing passwords of tags that have been singulated in the previous session. Assuming that tags are not highly mobile, most of the tags in the domain are singulated. Next, the reader can try passwords of tags that have left nearby domains, hoping that they have moved to this domain. This is only one example of what the back-end system can do. In other words, the back-end system collects information at the reader-tag level, fuses it to make system-level decisions, and implements the decisions by telling the readers which passwords to stream. In Sect. 4, we systematically consider increasingly strong attack scenarios, and how passwords can be chosen accordingly to defend against these attacks.

4 Attacks and Defenses

We vary our attack model by increasingly strengthening the attackers. In each case, we make specifications and/or modifications to our generalized scheme in Sect. 3 to defend against the progressively stronger attacks. Our defenses are focused on protecting against inventorying and tracking. We also consider how privacy can be traded off for efficiency in some of these cases, as determined by the application.

4.1 Eavesdropping

Eavesdropping occurs when an attacker listens to communications between readers and the respective tags in their domains. Assuming an attacker knows the dummy signal, it can collect passwords, and even differentiate which ones are stale (that is, have been updated) and which ones are still fresh. Note, however, that tag IDs are never exposed in any RFID channel, since they always remain in the back-end system and in readers. Thus, attackers can never succeed in inventorying via eavesdropping.

One-way Eavesdropping. Since tags are passive, readers broadcast at greater powers than tags backscatter. As a result, reader-to-tag channels are much more vulnerable to eavesdropping than tag-to-reader channels. Suppose only reader-to-tag eavesdropping is possible. Then, an attacker acquires only limited information to facilitate tracking. For example, if the attacker detects that the same password has been streamed by multiple readers in their respective domains over a relatively long period of time, and assuming passwords are not often reused, the attacker knows that the associated tag is likely not currently located inside any domain. This requires an attacker to eavesdrop for a relatively long period of time, move around to different domains, and/or enlist the help of other attackers, and communicate with them. Nonetheless, Sect. 2 assumes our system model has minimal communications and physical security that makes this unlikely to succeed.

Two-way Eavesdropping. If an attacker can eavesdrop on both reader-to-tag and tag-to-reader channels, it can easily track tags by following their respective password updates. With the help of other attackers, a tag can be followed as it moves from one domain to another. Even if an attacker misses a password update (as will likely occur due to our system model assumptions, which are essentially equivalent to the assumptions of the weakened attack model in [4]), and thus loses track of a tag, the attacker can quickly resume following it in the next singulation session of the reader corresponding to that tag's domain. To defend against this, we modify our proposed scheme in Sect. 3, by using the exclusive or (XOR) one-time pad algorithm in [4]. That is, a tag stores the vector $\Delta = \{\delta_1, \delta_2, \dots, \delta_m\}$, where m is a parameter that determines the eavesdropping resistance. The password of the tag is δ_1 . The back-end system also has a copy of Δ . When the reader streams the correct password, namely δ_1 , the tag generates another vector $\tilde{\Delta} = \{\tilde{\delta}_1, \tilde{\delta}_2, \dots, \tilde{\delta}_m\}$, and updates Δ locally as

$$\Delta := \{\delta_2 \oplus \tilde{\delta}_1, \delta_3 \oplus \tilde{\delta}_2, \dots, \delta_m \oplus \tilde{\delta}_{m-1}, 0 \oplus \tilde{\delta}_m\},$$

where \oplus is the exclusive or (XOR) operator. The tag then responds to the reader with $\tilde{\Delta}$, and the back-end system updates its copy of Δ . Note that in this algorithm, the updated password is not sent back to the reader. Instead, only the bit-wise difference is sent. Since memory is built into this scheme, and the one-time pad is information theoretically secure, the attacker has no information about the current password δ_1 , unless it had eavesdropped on all of the last m Δ updates of this tag, which is unlikely to have occurred as long as m is large enough. Singulation efficiency obviously degrades as m is increased, and is traded off for eavesdropping resistance, and thus, tracking resistance.

4.2 Tag Spoofing

Tag spoofing occurs when an attacker masquerades as a tag to a reader. For example, an attacker listens to a streamed password, and responds with a password update accordingly. If the streamed password matches with a tag's password, the combined responses of the tag and attacker quickly allow the reader to know the presence of the attacker. If the password does not match any tag's password, then the attacker gives the only non-dummy signal response, and the reader is fooled. As a result, there is a tag somewhere that no longer has a password synchronized with the back-end system. This creates an anomaly that can be used to notify the application to take the requisite action. That is, suppose the orphaned tag is in some domain, and the associated reader there begins a singulation session. The reader is not able to finish the session because the orphaned tag constantly responds with the dummy signal, since its password is not synchronized. A more active defense against tag spoofing is for tags to authenticate themselves to readers when they respond with updates. One method is to use separate dedicated passwords. Another way is to use password histories stored in the tags. That is, tags can authenticate themselves by responding with old passwords. The reader can also issue challenges for specific passwords in the tag's history. This idea of recycling passwords deserves further analysis.

Note that tag spoofing does not directly threaten privacy, since attackers are not inventorying or tracking tags. An attacker, however, can launch a tag spoofing attack as part of a combination attack, to compromise privacy.

4.3 Reader Spoofing

Reader spoofing occurs when an attacker masquerades as a reader to a tag. This is fundamentally a reader authentication issue. That is, given our defenses to eavesdropping as described in Sect. 4.1, attackers are unlikely to acquire passwords. As such, tags only respond with dummy signals to attacker queries, and therefore, no information is leaked. Nonetheless, suppose that the minimal level of communications and physical security mentioned in Sect. 2 is weakened (or equivalently, that the attackers are strengthened) to the level that eavesdropping allows attackers to acquire passwords. It is within this context that we provide defenses to reader spoofing to protect against tracking. Note that inventorying is not possible, for the same reason explained in Sect. 4.1.

Replaying Single Passwords. A tag is in great danger once an attacker acquires its password (through whatever means). That is, the attacker can immediately send the password to the tag, and the tag responds with a password update to the attacker. Note that in this case, we are assuming the attacker also knows where the tag is. Since the back-end system no longer has password synchronization with this tag, the tag is essentially lost. The attacker gains total control of the tag, and can thus track it. This applies even if we use the exclusive or (XOR) one-time pad algorithm. To defend against such an attack, we can use a throttling concept, similar to the ideas presented in [2], [4]. Recall, a tag does not respond to any queries when it is in sleep mode. When a tag goes into sleep mode after it has been singulated, a clock begins in the tag, and is also synchronized with a clock in the back-end system via the reader. The tag wakes up after the current singulation session is complete, or a throttling time period (possibly randomly and dynamically generated) elapses, whichever occurs later. Usually, the throttling time period is longer than the singulation session time. Assuming that the attacker knows the throttling time period, a simple strategy is for it to query the tag with its password immediately after it wakes up. As a defense, the back-end system can also schedule the reader to transmit that password at that same time. This does not prevent attacker-tag synchronization. But it does prevent reader-tag de-synchronization. We note that this defense is rather weak, and even difficult to implement, since it is not immediately clear how to schedule password streams to query tags at the correct times. Nonetheless, the attack provided here is rather strong, since acquiring both the password and location of a tag is extremely difficult, as explained in Sect. 4.1. In the following, we provide a more realistic reader spoofing attack.

Replaying Password Streams. Another means of spoofing readers is replaying password streams, again assuming that an attacker can eavesdrop. Note that in this case, we assume the attacker does not know where the tags associated with the passwords are located. There are several approaches the attacker can take. First, it

can replay a reader-to-tag password stream in the same domain. If the replay takes place shortly after the original stream, not much information can be garnered by the attacker. Passwords that had not matched in the original stream still do not match (assuming no new tags join the domain in that short time). Passwords that had matched are no longer valid. Therefore, this attack is rather weak. An alternative is that the attacker replay the stream in another domain (by sending the stream to another attacker, or by the attacker moving to the other domain). The idea is that non-matching passwords in the original stream might belong to tags in other domains. This attack, however, is again rather weak. For example, the readers can dynamically keep track of where tags are with the help of the back-end system. As long as tags are not highly mobile, a reader only potentially streams a non-matching password when a tag has left its domain (and has not yet been singulated in another domain), or when a new tag has entered its domain. As well, by the time the attacker (or its affiliate) replays a potentially matching password in another domain, it is likely that the tag that just arrived already has had its password updated, thus rendering the attacker's password stale. A more aggressive defense is for a reader to purposely inject garbage intermittently in its password streams. This obviously lengthens singulation times, but the added privacy is that an attacker replaying such a stream is slowed down, causing its passwords to become stale before they can be used, reminiscent of throttling. Note that garbage injection does not help if the stream is replayed (by the attacker's affiliate) in real-time. A more offensive defense strategy is for a reader to inject stale passwords, instead of, or in addition to garbage. Tags are allotted additional memory to hold their password histories. A tag that receives queries containing stale passwords, records this information, and relays it to its reader. This information is collected, processed in the back-end database, and the results are reported to the application, which can take further action to capture the attackers. Of course, a counter-attack is for attackers to record their eavesdropping histories. Note, however, that despite our strengthened attackers in this section, their combined abilities are still assumed to be weaker than the centralized processing power of the back-end system. Thus, the counter-attack is not very effective, for example, if the attackers' combined history memory is smaller than that of the back-end system.

Second, the attacker can replay the tag-to-reader stream. At first glance, this might seem quite troubling since the passwords are fresh. But the passwords from this stream are useless in other domains. If the stream is replayed in the same domain, then we are back to the situation of replaying single passwords.

Finally, replaying a combination of both the reader-to-tag and tag-to-reader streams, and even the streams of multiple domains requires detailed analysis which is beyond the scope of this paper (but is very much the subject of future research).

4.4 Physically Attacking Tags or Readers

We discuss the possibility of attackers physically capturing tags and readers (temporarily or permanently), and thus having unfettered access to their memories. This certainly is well-beyond the capabilities of a realistic attacker. But we offer a discussion to illustrate the inherent privacy aspects of our proposed scheme.

Our proposed scheme can still function if tag IDs are absent in tags and readers. If we implement our scheme in such a way, inventorying is unsuccessful even if tags or readers are captured. Nonetheless, if a reader and the back-end system are constrained in their communications with each other, it may be more efficient to have some tag ID memory in readers. Therefore, there is a privacy tradeoff. In the unlikely event that a reader is captured by an attacker, it gains tag ID information, and can perform inventorying. Conversely, tracking is readily achieved. An attacker can capture a tag, password-synchronize with it, and then replace it in the physical space. The tag is now totally controlled by the attacker to perform tracking.

We do not consider physical tag or reader spoofing attacks, where a malicious tag or reader (masquerading as legitimate) is placed inside our physical space. This is a natural attack that follows after capturing a tag or reader.

5 RFID Supporting Infrastructure

Using our proposed scheme as a foundation, a much richer privacy system can be explored. One aspect is offering the application a dynamic risk assessment of the system, which it can use to make real-time decisions. For example, the back-end system can keep histories of the time lengths between password updates, as well as the locations of tags. These are just two sets of temporal and spatial data. Other data can also be easily collected and processed to evaluate the risks of individual parts of the system, as well as the entire system as a whole. For example, a risk value can be dynamically assigned to tags, to indicate their likelihoods of having already been compromised by attackers. Readers then can interact with the tags, based on these tiered risk levels, depending on the requirements of the application.

Another aspect is optimizing the performance of our system. That is, given certain constraints, such as an attack model, what is the maximum level of privacy our system can provide, and what are the algorithms to achieve this? The optimization can be global, with the back-end system operating as the centralized controller. The optimization can also be local, which models a situation where readers have constrained access to the back-end system, and thus, do not have global knowledge of the system.

These ideas are largely related to the implementation of the back-end system, otherwise known as the supporting infrastructure of an RFID system. The research in the relatively young area of RFID has been primarily focused on the interactions between readers and tags. The next frontier is understanding the supporting infrastructure, and in particular, how privacy can be implemented at this level.

6 Conclusion

In this paper, we propose a novel privacy-protecting RFID system using *password streaming*. Our proposed scheme assumes a reader knows the presence

of multiple tags, in its transmit range, and immediately starts streaming tag passwords associated with tag IDs, in a broadcast fashion, allowing for tags to respond in between passwords. A centralized back-end system coordinates the password streaming of multiple readers in their respective domains. Our proposed scheme benefits from being efficient and light-weight, which is achieved by pushing complexity to the back-end system. Yet, it still defends against strong attack scenarios.

Future work includes exploring our proposed scheme in detail. This includes investigating the various components of our system, and how they can be used to cooperatively defend against different types of attacks. As well, the role of RFID supporting infrastructure in protecting privacy is not immediately clear, and deserves further work.

References

1. Juels, A.: RFID security and privacy: A research survey. *IEEE J. Sel. Areas Commun.* 24, 381–394 (2006)
2. Langheinrich, M., Marti, R.: Practical minimalist cryptography for RFID privacy. *IEEE Systems Journal* 1, 115–128 (2007)
3. Shamir, A.: How to share a secret. *Comm. of the ACM* 22, 612–613 (1979)
4. Juels, A.: Minimalist cryptography for low-cost RFID tags. In: *Security in Communication Networks*, Amalfi, Italy, pp. 149–164 (September 2004)
5. Law, C., Lee, K., Siu, K.-Y.: Efficient memoryless protocol for tag identification. In: *Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, Boston, MA, August 2000, pp. 75–84 (2000)
6. Vogt, H.: Multiple object identification with passive RFID tags. In: *International Conference on Systems, Man and Cybernetics*, Hammamet, Tunisia (October 2002)

RDDS 2008 PC Co-chairs' Message

Middleware has become a popular technology for building distributed systems from sensor networks to large-scale peer-to-peer (P2P) networks. Support such as asynchronous and multipoint communication is well suited for constructing reactive distributed computing applications over wired and wireless networks environments. While the middleware infrastructures exhibit attractive features from an application-development perspective (e.g., portability, interoperability, adaptability), they are often lacking in robustness and reliability. This workshop focuses on reliable decentralized distributed systems. While decentralized architectures are gaining popularity in most application domains, there is still some reluctance in deploying them in systems with high dependability requirements. Due to their increasing size and complexity, such systems compound many reliability problems that necessitate different strategies and solutions. This has led, over the past few years, to several academic and industrial research efforts aimed at correcting this deficiency. The aim of the RDDS Workshop is to bring researchers and practitioners together, to further our insights into reliable decentralized architectures and to investigate collectively the challenges that remain. The program for RDDS 2008 consisted of five research papers of high quality, covering diverse topics. Each paper was reviewed by at least 3 reviewers. We are very grateful to the members of the RDDS 2008 Technical Program Committee for helping us to assemble such an outstanding program. We would like to express our deep appreciation to the authors for submitting publications of such high quality, and for sharing the results of their research work with the rest of the community.

November 2008

Achour Mostefaoui
Eiko Yoneki

Reliable Peer-to-Peer Semantic Knowledge Sharing System

Abdul-Rahman Mawlood-Yunis

School of Computer Science, Carleton University
1125 Colonel By Drive, Ottawa, Ontario, K1S5B6, Canada
armyunis@scs.carleton.ca

Abstract. Reliability issues arise in Peer-to-Peer Semantic Knowledge Sharing (P2PSKS) systems. P2PSKS systems are prone to disconnection failures. Disconnection failures deprive the network peers from having access to complete knowledge in the P2PSKS system. Disconnection failures arise when the P2PSKS systems employ the adaptive query routing methods in order to avoid flooding the networks with redundant search messages or queries during information exchange. P2PSKS systems need to utilize fault-tolerant query result evaluation function in order to be reliable. The current publication will focus on two objectives in P2PSKS research: 1. reliability problem identification and 2. reliability problem solutions. The P2PSKS system reliability problem will be identified through the use of simulation modeling techniques. The P2PSKS system reliability problem solutions will be advanced through adaptations of the generous tit-for-tat method which was originally developed in evolutionary game theory. The current research will demonstrate more reliable solutions to address P2PSKS system reliability issues by advancing the Fault-Tolerant Adaptive Query Routing (FTAQR) algorithm.

1 Introduction

Current research directions in Peer-to-Peer Semantic Knowledge Sharing (P2PSKS) systems are evolving to combine two complementary technologies: 1. Peer-to-Peer (P2P) networks and 2. formally-structured information (ontology). P2PSKS systems represent the next step in the evolution of P2P networks because P2PSKS systems incorporate several additional features not present in P2P networks. The most important additional feature found in P2PSKS systems is the point-to-point mapping. The point-to-point mapping is used as a translational capability to forward queries between peers under the conditions when the peers possess different data schema or knowledge representations. P2P networks which employ ontologies include four examples: 1. P2P knowledge management systems [5,9,15,16,8], 2. P2P Semantic Web [10,18,19], 3. P2P Web Services [4] and 4. P2P systems which require cooperation among distributed and autonomous peers with heterogeneous information sources, i.e. local ontologies [16].

Ontologies are advantageous in P2P networks because they provide the possibility for improving search and content retrieval. The incorporation of ontologies

in P2P networks has been previously reported in the scientific literature in three research precedents: 1. creation of semantic networks on existing P2P networks which has been referred to as semantic overlay networks, 2. semantic-based query routing and 3. adaptive query routing. These three methods are used to avoid flooding the P2PSKS systems with redundant search messages or queries during information exchange. Flood avoidance refers to selectively sending queries to only those peers which are highly knowledgeable or expert about query content.

The current research will focus on P2PSKS systems reliability issues. The P2PSKS systems reliability refers to the systems resiliency under the conditions for handling temporary faults. P2PSKS systems need to utilize fault-tolerant query result evaluation function in order to be reliable. That is, proper query result evolution function is absolute necessary to guide query forwarding or risk network connectivity deterioration.

In order to address the P2PSKS systems reliability issues, we have invented the Fault-Tolerant Adaptive Query Routing (FTAQR) algorithm. The FTAQR algorithm is capable of handling temporary faults under three conditions: 1. systems unavailability, 2. peer misbehavior, and 3. mapping corruption. The further details of the three conditions will be provided below. The FTAQR algorithm is based on adaptations of the generous tit-for-tat method which was originally developed in evolutionary game theory (see e.g. [2] for information on iterative prisoner's dilemma).

The importance of this study goes beyond the P2PSKS systems. Fault-tolerant adaptive query routing is essential in other research directions as well. These include, for example, automatic consensus building or bottom-up construction of ontology, and adequate resource utilization in large and open information system. This is because these researches also depend on the correct decision about query routing adaptation.

The remainder of this publication is organized into the following sections: Section 2 discusses faults due to resource unavailability and peers misbehavior. Section 3 presents the FTAQR Algorithm and its related design decisions. Section 4 discusses simulation building blocks and simulation setup. Section 5 presents initial result evaluation and reliability metrics. Section 6 reviews related work, and finally Section 7 concludes the paper and identifies directions for future work.

2 Resource Unavailability and Peers Misbehavior

Formal fault definition and fault classification along temporal dimension as well the effects of mapping corruption on bottom-up construction of ontology have been provided in [13,14]. Hence, we avoid repeating this issues here. However, for the sake of self-containment, we provide a short description of the reasons for temporary faults due to resource or service unavailability and peers misbehavior in this section:

- a) It has been pointed out by Gal [7] that the design of the conceptual schema for information services possesses special properties. These include (i) a rapid

change of data sources and metadata; and (ii) instability, since there is no control over the information sources. The *availability* of information sources is solely dependent upon information source providers. A possible scenario is the temporary unavailability of information when such information is needed. This possibility is particularly acute during query execution.

- b) System operation in P2PSKS systems depends on the honest conduct of peers. A peer could be dishonest or biased in its interaction with other peers during query forwarding for reasons such as selfishness or greed. There are various ways through which a peer could influence query forwarding. These include (i) not forwarding a query to other peers during transitive query process (ii) not forwarding answers to the other peers; or (iii) altering or delaying queries (results) before forwarding them to other peers.

In both above cases, the querying peer will receive incorrect query results, i.e. faults occur. These faults could be permanent or temporary. Hence, it is desirable for P2PSKS systems to be able to differentiate between different types of faults (permanent, intermittent and transient), and tolerate the non-permanent ones (transient and intermittent). This is because P2PSKS systems employing adaptive query routing method risk, unnecessarily, partial or total disconnection, based on the network topology at the time, if they do not tolerate temporary faults. More detailed discussion about this observation is provided in [12]

3 FTAQR Algorithm Description and Design Decisions

In this section we are going to describe algorithm steps and procedure. The algorithm is simple in concept, easy to implement and highly effective. Several design decision have been made during algorithm development. These include:

- A) **Use of local knowledge.** Peers have only knowledge about their immediate neighbors. Peer's knowledge is about their belief in the reliability or ability of their neighboring peers on providing correct answers to their queries. The reliability value ≥ 1 refers to total reliability, and reliability value ≤ 0 refers to totally unreliability . Peers disconnected from each other when reliability values reaches ≤ 0 .
- B) **Normalization is not applied.** Sending query on an outgoing link could result in several query answers. The number of query answers depends on the number of cycles in the network starting from the querying peer. All answers are treated equally. That is, no extra weight is given to any particular answer or querying path.
- C) **Use of average value.** The average value of query answers is used for evaluating the reliability of the outgoing links.

These decisions are made to make the algorithm simple to be understood. Future revision of these decisions is possible. The algorithm is made up of three essential functions: 1. initialization, 2. result evaluation, and 3. an update function, and proceeds along the following steps:

1. At the startup of network, peers start *connection*; connected peers set their trust value in each other to 1, and system parameters for query result evaluation are initialized.
2. query result evaluation checks for the ($\equiv, \supset, \subset, *, \perp$) relations between concepts in query answer and concepts in querying peer's local data set. These relations imply that [100%, 75%, 75%, 25%, 0%] of concepts returned by query answer are understood by querying peer respectively. That is, the relation between query concepts and peers' local concepts are **exact same, related, and totally not related**.
3. based on query result evaluation relation, i.e. step 2, peers update their confidence in the reliability of their outgoing links. The numerical values used for updating are [0.2, 0.1, 0.1, 0.05, -0.2]. The update values correspond to the semantic relation between query answer concepts and peer's local concepts.

We decrease peer's confidence in its outgoing link by 0.2 every time it receives incorrect query answers. This is in order to initially tolerate up to 4 faults in sequence. Hence, the generosity part of algorithm to tolerate temporary faults. After that, peers will be treated by the tit-for-tat rule. That is, peers will be punished for returning any incorrect query answer, and rewarded for their correct query answers. Snapshot pseudocodes corresponding to the three described steps are as follow:

```

Initialize () { ChangeInStrength [ ]  $\leftarrow$  {0.2, 0.1, 0.1, 0.05, -0.2}
    While(aPeer.haveMoreLinks())
        aPeer.setOutGoingMappingLink  $\leftarrow$  1
    End While
}

Result_Evaluation() {
    result  $\leftarrow$  query.getResultContent()
    newStrength  $\leftarrow$  0
    IF (Relation_is_synonyms (result))
        newStrength  $\leftarrow$  anEdge.getStrength() + ChangeInStrength[0]
    Else If (Relation_is_hypernyms (result) )
        newStrength  $\leftarrow$  anEdge.getStrength() + ChangeInStrength[1]
    Else If (Relation_is_hyponyms (result))
        newStrength  $\leftarrow$  anEdge.getStrength() + ChangeInStrength[2]
    Else If (Related (result))
        newStrength  $\leftarrow$  anEdge.getStrength() + ChangeInStrength[3]
    Else
        newStrength  $\leftarrow$  anEdge.getStrength() + ChangeInStrength[4]
    End IF
}

Update (int newStrength) {
    anEdge.setStrength(newStrength)
    removeUnusedLinks (anEdge)
}

```

4 Simulation Building Blocks and Setup

In this section, the simulation building blocks are described. These include a short description of peers, resources, semantic neighborhood and query formulator abstracts.

4.1 Peers

A peer represents an instance of a participant in the network. Each peer has a unique id, `peerID`, data schema, and `schema_description` or profile. The latter is used for forming semantic neighborhood.

4.2 Resource

To simulate the generic characteristics of a P2PSKS system and keep distance to any particular realization, we abstract in our simulation from the underlining data model and consider data to be set of concepts. A file which contains Laptop Ontology developed by Stanford University¹ constitute peer's data/knowledge. Laptop ontology is made of 40 concepts, each peer stores a percentage of the total ontology concepts (file content). Network peers are divided into four groups automatically. The number of peers in each group is $N/4$, where N is the total number of peers. The size of the dataset each peer holds is based on the group type it belongs to. Group1, Group2, Group3 and Group4 hold 15%, 30%, 50% and 70% of total concepts respectively. Peers are assigned to the groups randomly.

4.3 Semantic Neighborhood

In our simulation, we employ the discovery method. Peer schema descriptions (profile file) are simple XML files. On the network startup, peers exchange their profile files and a tree comparison algorithm is used to determine the similarity relation between schemas. When a new peer joins the network, it will advertise its schema description, and peers with similar schema will reply to the advertisement and connection is established. Beside the similarity function (*sim*) which has to be greater than a predefined threshold (e.g. $sim \geq \delta$) in order for peers to be able to connect, the number of connections each peer could have is also another restriction. The in/out connection degree(I, O), and δ are user defined variables and their values are defined on the simulation startup. The following snapshot pseudocode represents the described steps.

```

/*compare, a tree comparison function call */
Set sourceAndTarget ← new HashSet()
For( int i ← 0 To i< numberOfNodes - 1)
  For ( int j← i To j< numberOfNodes - 1)
    /* SchemaCollection contains all schemas */
    compare (sourceAndTarget, SchemaCollection[i], SchemaCollection[j +1 ], i, j+1)

```

¹ <http://www.ksl.stanford.edu/DAML/laptops.owl>

```

        j ← j+1
    End For
    i ← i+1
End For
/*set of function calls to rank neighboring peers and create semantic neighborhood created */
connectionMap (sourceAndTarget )
sortConnectionMap (connections, rows, columns)
sortByStrength (sortedMapById )
semanticNeighborhood (getGroups ())

```

4.4 Query Formulator

In our P2P-SKS system simulation, queries are made up of 4 concepts and SELECT query operator is used. Query concepts are chosen from local data set, and any peer could initiate a query. Peers that initiate queries as well as query concepts are selected randomly. At each run a new query is created and a new initiator is selected. Chances for the new query concepts and query initiators to be different than prior concepts and initiators are high. The probability of a peer to be a query initiator or a querier at each run of our automatic query formulator is $1/N$, where N is the number of peers of the simulation. The probability of the query to be exactly the same query as the previous one is $(1/D)^{\|f\|}$, where D is the size of data set each peer possesses, and $\|f\|$ is number of concepts that comprise the query content. The probability of the same peer to pose the the same query in two different system runs is then the multiplication of both above terms, i.e., $(1/N)(1/D)^{\|f\|}$. The following snapshot pseudocode represents the described steps.

```

/* Queries variable is peers local data set */
ArrayList queries ← FileInput.read()
Vector queryConcepts
Query q ← null
numberOfConceptsInQuery ← 4
For(int i ← 0 To i < numberOfConceptsInQuery) {
    int QueryIndex ← Random.uniform.nextIntFromTo(0, queries.size() -1)
    queryConcepts.add(queries.get(QueryIndex).toString())
    i ← i+1
End For
/* create query object with its content */
q ← new Query (queryConcepts)

```

The reader may notice that, two other essential components, **mapping** and **router**, are not described here. The former is not used because in our current simulation version, although different in sizes, all peers use same resource, and the latter is simulated only partially through simple query forwarding and connection dropping. Peers use the semantic neighborhood initially created for search and content retrieval, and drop relations with related neighbors when their strength

level, i.e., ability to provide correct query answers, reach zero. Further improvement regarding the mapping and router issue will be considered in subsequent works.

5 Experimental Results and Reliability Metrics

In this section we describe network **reliability metric**, **fault simulation** and **experimental results**. Connectivity is used as a **metric** for evaluating network reliability. That is, we observe the network deterioration speed (slop function), and the network connectivity degree of the system setups running for a constant period of time. The experiments run under two different settings: with and without the capability to tolerate faults. The variation in the network deterioration trend and the connectivity degree between the two system settings is reported as the difference in the reliability that a FTAQR algorithm guarantee.

Faults are generated using the knowledge about peers' different data sizes, query formulation and routing strategy. That is, when a query is created with a set of peer's local concepts and that query is passed through other peers which have only a subset of query concept constituents, then the end query result will contain only the minimum common denominator of concepts. Hence the fault is generated.

Figures (1, 3) and (2, 4) represent initial results for two different experimental settings. The system components, i.e. data sets, query formulator, query router and number of peers (15), are same for both settings. The only difference between the two system settings is the difference in query evaluation function, system fault-tolerance capability. The results clearly demonstrate that building P2PSKS systems with built-in fault-tolerance capability increases system reliability. The network deterioration trend for the second system setting, Figure 4, is less sharp than the network deterioration trend in the Figure 3. Similarly, the difference between network connection states demonstrate that a P2PSKS system with fault-tolerant capability is more reliable than one without such a capability. Figures 1 and 2 show that after running system simulation for the period of $4 * 10^2$ ms, the network disconnection rate for the first system setting was $> 93\%$, and only 33% for the second one. Table 5 summarize these results.

Its worth to mention that, in both system settings the network could reach to the state of total disconnection. This is because, in the current version of our simulation, we do not add new peers to the system during system operation. That is, while peers lose connection because of incorrect query results, they do not make new connections. Thus, the possibility of total disconnection arises. Similarly, the disconnected peers are not considered for reconnection. Further, in situations where peers are about to be totally disconnected, e.g. when peers have only one outgoing link, a different action from the query result evaluation component might be necessary. This includes, for example, changing the way peers are punished for the incorrect query answers. These are issues which we will consider in the subsequent simulation versions.

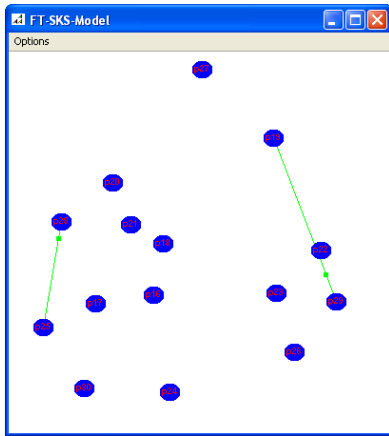


Fig. 1. Network disconnection when FTAQR is Not used

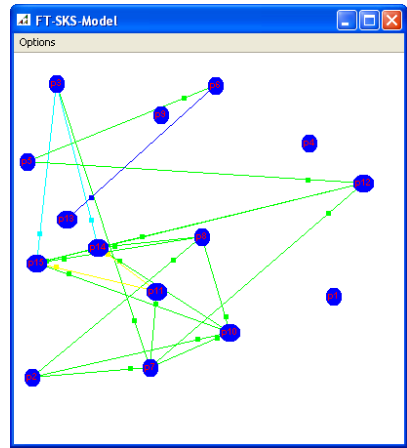


Fig. 2. Network disconnection when FTAQR is used

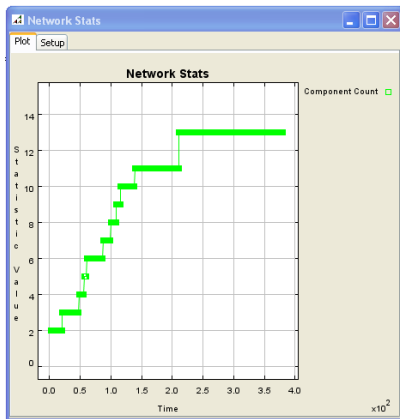


Fig. 3. Network determination when FTAQR is Not used

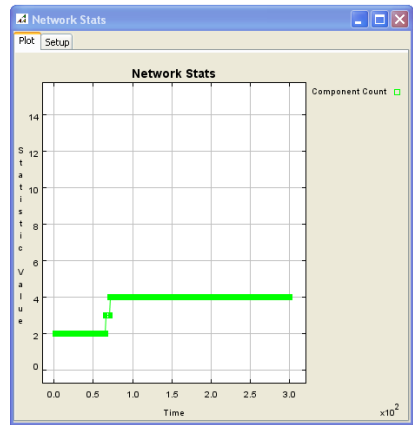


Fig. 4. Network determination when FTAQR is used

Table 1. FTAQR's Effect on Network Connection and Network Deterioration Trend

	P2PSKS without FTAQR	P2P with FTAQR
Network Disconnection	93%	33%
Function Slop	3	1.25

6 Related Work

Loser et al. [11] suggest that the information evaluation strategy, among others, is a criterion for distinguishing adaptive query routing in semantic overlay networks from each other. While, we concur with Loser, we also presented a new algorithm for query result evaluation. Acknowledging the fact that P2PSKS networks could change during query propagation, i.e., a peer may become temporarily unavailable or a new peer with relevant information source joins the network, Zaihrayeu [20] highlights three different scenarios which have the potential for generating faults. Zaihrayeu, however, tries to transform (avoid) the identified problems through a set of assumptions. Our approach is resilient to temporally unavailability problem highlight by Zaihrayeu. More generally, our approach is resilient to the update problems associated with P2P networks. Other relevant contributions such as [9] and [15] do not consider fault-tolerance. Hence, our FTAQR algorithm complements these contributions. Checking on recurrent incorrect query answers by semantically relevant peers in [8], and probabilistic based fault-tolerance approach employed by [1] could be further improved using our FTAQR algorithm through tolerating non-permanent fault.

7 Conclusion and Future Work

In this work, we identified that reliability is an issue in P2PSKS systems. P2PSKS system referred to combining two different technology worlds: standard P2P networks and Ontologies. The identification of the problem was demonstrated through the possibility of total or partial network failure due to the existence of faults in the query evaluation function. The occurrence of faults was presented through two different scenarios: temporally unavailability of peers and peers misbehavior. In addition to reliability identification problem in P2PSKS systems, a solution to the problem was proposed as well. The proposed solution, FTARQ algorithm, did not require time, information or component redundancy. FTARQ algorithm is an adaptation of generosity tit-for-tat technique used in game theory. The effectiveness of the FTARQ algorithm demonstrated through system simulation. The description of simulation components, and test results were provided as well.

As a future work, we are working on extending our current FTAQR algorithm as follow: Instead of allowing all query results to equally influence the confidence peers have in their out-going links, we could use various majority voting techniques to derive new algorithm. The voting algorithm would be compared to the current FTAQR algorithm to further study system reliability improvement. We are also working on building a more complete P2PSKS system simulation than the current one. A completed system simulation would be used for studying various aspects and issues of P2PSKS systems. This includes further examination of the influence that an FTAQR algorithm could have on systems such as Chatty Web [1], KEx [5], Piazza [9], P2PSLN [8].

References

1. Aberer, K., Cudre-Mauroux, P., Hauswirth, M.: Start making sense: The Chatty Web approach for global semantic agreements. *Journal of Web Semantics* 1(1), 89–114 (2003)
2. Axelrod, R.: *The Complexity of Cooperation*. Princeton University Press, Princeton (1997)
3. Lyu, M.R.: *Software Fault Tolerance*. Wiley Publishing, Chichester (1995)
4. Bianchini, D., De Antonellis, V., Melchiori, M., Salvi, D., Bianchini, D.: Peer-to-peer semantic-based web service discovery: state of the art, Technical Report, Dipartimento di Elettronica per l'Automazione Universit di (2006)
5. Bonifacio, M., Bouquet, P., Mameli, G., Nori: Peer-mediated distributed knowledge management. In: van Elst, L., Dignum, V., Abecker, A. (eds.) *AMKM 2003. LNCS (LNAI)*, vol. 2926, pp. 31–47. Springer, Heidelberg (2004)
6. Fergus, P., Mingkhwan, A., Merabti, M., Hanneghan, M.: Distributed emergent semantics in P2P networks. In: *Proc. of the Second IASTED International Conference on Information and Knowledge Sharing*, pp. 75–82 (2003)
7. Gal, A.: Semantic Interoperability in Information Services: Experiencing with CoopWARE. *SIGMOD Record* 28(1), 68–75 (1999)
8. Hai, Z., Jie, L., et al.: Query Routing in a Peer-to-Peer Semantic Link Network. *Computational Intelligence* 21(2), 197–216 (2005)
9. Halevy, A., Ives, Z., Mork, P., Tatarinov, I.: Piazza: Mediation and integration infrastructure for semantic web data. In: *proceedings of the International World-Wide Web Conference WWW 2003* (2003)
10. Haase, P., Broekstra, J., et al.: Bibster – A Semantics-based Bibliographic Peer-to-Peer System. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) *ISWC 2004*, vol. 3298, pp. 122–136. Springer, Heidelberg (2004)
11. Loser, A., Staab, S., Tempich, C.: Semantic social overlay networks. *IEEE Journal on Selected Areas in Communications* 25(1), 5–14 (2007)
12. Mawlood-Yunis, A.-R., Weiss, M., Santoro, N.: Issues for Robust Consensus Building in P2P Networks. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops. LNCS*, vol. 4278, pp. 1020–1028. Springer, Heidelberg (2006)
13. Mawlood-Yunis, A.-R., Weiss, M., Santoro, N.: Fault Classification in P2P Semantic Mapping. In: *Workshop on Semantic Web for Collaborative Knowledge Acquisition (SWeCKa) at Intl. Conf. on Artificial Intelligence, IJCAI* (2007)
14. Mawlood-Yunis, A.-R., Weiss, M., Santoro, N.: Fault-tolerant Emergent Semantics in P2P Networks. In: *Semantic Web Engineering in the Knowledge Society*. Idea Group (to appear, 2008)
15. Mena, E., Illarramendi, A., et al.: OBSERVER: an approach for query processing in global information systems based on interpretation across pre-existing ontologies. *Distributed and Parallel Databases* 8(2), 223–271 (2000)
16. Kementsietsidis, A., Arenas, M., et al.: Managing Data Mappings in the Hyperion Project. In: *The 19th Intl. Conf. on Data Engineering (ICDE)*, pp. 732–734 (2003)
17. Paradhan, D.K.: *Fault-Tolerant Computing System Design*. Prentice-Hall PTR publication, Englewood Cliffs (1996)
18. Rousset, M., Chatalic, P., et al.: Somewhere in the Semantic Web. In: *Intl. Workshop on Principles and Practice of Semantic Web Reasoning*, pp. 84–99 (2006)
19. Staab, S., Stuckenschmidt, S.: *Semantic Web and Peer-to-Peer*. Springer, Heidelberg (2006)
20. Zaihrayeu, I.: *Towards Peer-to-Peer Information Management Systems*. PhD Dissertation, International Doctorate School in Information and Communication Technologies, DIT - University of Trento (2006)

How to Improve the Reliability of Chord?*

Jacek Cichoń¹, Andrzej Jasiński², Rafał Kapelko¹, and Marcin Zawada¹

¹ Institute of Mathematics and Computer Science,
Wrocław University of Technology, Poland

² Institute of Mathematics and Computer Science, Opole University, Poland

Abstract. In this paper we focus on Chord P2P protocol and we study the process of unexpected departures of nodes from this system. Each of such departures may effect in losing any information and in classical versions of this protocol the probability of losing some information is proportional to the quantity of information put into this system.

This effect can be partially solved by gathering in the protocol multiple copies (replicas) of information. The replication mechanism was proposed by many authors. We present a detailed analysis of one variant of blind replication and show that this solution only partially solves the problem. Next we propose two less obvious modifications of the Chord protocol. We call the first construction a *direct sums of Chords* and the second - a *folded Chord*. We discuss the recovery mechanisms of partially lost information in each of these systems and investigate their reliability. We show that our modification increases essentially the expected lifetime of information put into the system.

Our modifications of the Chord protocol are very soft and require only a small interference in the programming code of the original Chord protocol.

1 Introduction

We assume that the reader knows the classical Chord protocol (see [1] and [2]), which may be described as a structure

$$\text{Chord} = (\{0, 1\}^{160}, H, H_1) ,$$

where H is a hash function assigning position to each node and H_1 is a hash function assigning position of descriptors of documents. The space $\Omega = \{0, 1\}^{160}$ is identified with the set $\{0, 1, \dots, 2^{160} - 1\}$ considered as circular space with the ordering $0 < 1 < \dots < 2^{160} - 1 < 0 < \dots$. Each new node X obtains a position $H(Id)$ (where Id is an identifier of the node) in the space Ω and is responsible for the interval starting at point $H(Id)$ and ending at the next point from the set $\{H(Id') : Id' \neq Id\}$. This node is called the successor of the node X . Each

* Supported by the EU within the 6th Framework Programme under contract 001907 (DELIS) and by the grant No 331540 of the Institute of Mathematics and Computer Science, Wrocław University of Technology, Poland.

document with a descriptor doc is placed at point $H_1(doc)$ in the space Ω and the information about this document is stored by the node which is responsible for the interval into which $H_1(doc)$ falls.

Nodes arrive into the system, stay there for some period of time, and later they leave the system. During their lifetime in the system they gather some information in their local database. At the end of their lifetime in the system they may upload gathered information into the system (for example by sending it to the node's predecessor) or they may leave the system without uploading gathered information into the system. We call the first method a *regular departure* and we call one an *unexpected departure*. Note that in the classical Chord after an unexpected departure the whole information gathered during node's lifetime is lost (see e.g. [3]).

The goal of this paper is to show and analyze several modifications of the original Chord protocol which improve its reliability with respect to survival of information gathered by the system. During our investigations we considered only such modifications of the Chord protocol which neither destroy its logical clearness nor change its basic algorithms.

The paper is organized in such a way that in Section 2 we introduce three modifications of Chord and investigate its combinatorial and probabilistic properties. In Section 3 we discuss the mechanism of recovering of partially lost information from systems.

We shall identify the space Ω with the unit interval $[0, 1)$ and we shall interpret positions of nodes of a Chord as elements of $[0, 1)$. In other words, we identify the key space with a circle with unit circumference. Moreover, we may assume that one node is at point 0. The random sets corresponding to nodes of Chord are generated in the following way: we generate independently n random points X_1, \dots, X_n from $[0, 1)$ using the uniform distribution on $[0, 1)$ and then we sort them in increasing order and obtain a sequence $X_{1:n} \leq \dots \leq X_{n:n}$. This construction will be used as a formal model of the Chord protocol with $n + 1$ nodes. We call the segment $[X_{i:n}, X_{i+1:n})$ the interval controlled by the node i .

By $\Gamma(z)$ we denote the standard generalization of the factorial function. We will also use several times the Euler function $B(a, b) = \int_0^1 x^{a-1}(1-x)^{b-1} dx$ which is defined for all complex numbers a, b such that $\Re(a) > 0$ and $\Re(b) > 0$. Let us recall the following basic identity: $B(a, b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$.

Let X be a random variable. We denote its expected values by $\mathbf{E}[X]$. By **w.h.p.** (with high probability) we understand in this paper that the probability of considered events tends to 1 when the parameter n , denoting usually the number of nodes, tends to infinity.

We shall not give proofs of theorems in this paper - we will create a technical report with proofs and additional results and it will be available on web pages of the Institute of Mathematics and Computer Science, of the Wrocław University of Technology (Poland). Let us only mention that we use in proofs the technology of ordered statistics, asymptotic properties of the Beta function and the technology of analytic combinatorics.

2 Modifications of Chord

In this section we define and investigate three modifications of the classical Chord protocol and estimate the probability of not losing any information from the system after a simultaneous unexpected departure of a big group of nodes.

2.1 Chord with Replicas of Documents

The *Chord with L-replicas* of documents is the structure

$$r_L\text{-Chord} = (\Omega, H, \{H_1, \dots, H_L\}) ,$$

where H, H_1, \dots, H_L are independent hash functions, H is used for putting nodes into the system and H_1, \dots, H_L are used for placing documents into the system. We assume that each document doc is placed into each of the positions $H_1(doc), \dots, H_L(doc)$. Therefore, if we put d documents into the structure $r_L\text{-Chord}$ then the total number of „information items” in the system is approximately $d \cdot L$. We add one additional procedure to the $r_L\text{-Chord}$ protocol. Namely, we assume that each node periodically, with some fixed period Δ , checks whether all remaining copies of its documents are in the system.

Our strategy of replication of documents in Chord may be called a *blind strategy*. More advanced methods were investigated by many authors (see e.g. [4]), but we analyze in this paper only the simplest one - based on the family of independent hash functions.

Let $Z_{d:L:n}$ denote the number of nodes which must be removed in order to lose some information from the system. Then it can be proved that

$$\Pr[Z_{d:L:n} > k] = \frac{1}{B(k, n - k + 1)} \int_0^1 (1 - x^L)^d x^{k-1} (1 - x)^{n-k} dx ,$$

and from this equation we can derive the following theorem:

Theorem 1. *Let the number of nodes in $r_L\text{-Chord}$ be $n + 1$, let d denote the number of documents put into this structure. Let $Z_{d:L:n}$ denote the number of nodes which must be removed in order to lose some information from the system. Then*

$$\mathbf{E}[Z_{d:L:n}] = 1 + n \cdot \frac{\Gamma(d + 1)\Gamma(1 + \frac{1}{L})}{\Gamma(d + 1 + \frac{1}{L})}$$

Let us assume that $d = \delta \cdot n$ i.e. that δ is the mean number of documents put into the system by each node. In real application, when the number of nodes is large, we may suspect that $1 \leq \delta \leq 100$. From Theorem 1 we may deduce the following result:

Corollary 1. $\mathbf{E}[Z_{\delta n:L:n}] = 1 + \frac{\Gamma(1 + \frac{1}{L})}{\delta^{\frac{1}{L}}} n^{1 - \frac{1}{L}} + O(n^{-\frac{1}{L}}) .$

In the classical case $L = 1$, so $\mathbf{E}[Z_{\delta n:1:n}] \approx 1 + \frac{1}{\delta}$ and this means that if $\delta \geq 1$ then after each unexpected departure some information will disappear from

the system with high probability. The situation changes when we keep in the Chord two copies of each document, namely if $L = 2$ then we have $\mathbf{E}[Z_{\delta n:2:n}] \approx \Gamma(\frac{3}{2})\sqrt{\frac{\pi}{\delta}} \approx 0.89\sqrt{\frac{\pi}{\delta}}$.

Observe that the expected value of the random variable $Z_{\delta n:L:n}$ depends on the number of documents.

We say that a subset A of nodes is **safe** if after removing nodes from the set A no information will completely disappear from the system. We say that a subset A of nodes is **unsafe** if after removing nodes from A some information will completely disappear from the system.

Theorem 2. *Let n be the number of nodes in r_2 -Chord with δn documents. Suppose that $1 \leq \delta \leq 1000$. Let A be a subset of the set of nodes.*

1. *If $|A| \leq \frac{1}{\ln n} \sqrt{\frac{\pi}{\delta}}$ then A is safe w.h.p.*
2. *If $|A| \geq \sqrt{3 \ln n} \sqrt{\frac{\pi}{\delta}}$ then A is unsafe w.h.p.*

We can generalize the above Theorem for r_L -Chord for arbitrary fixed $L \geq 2$.

2.2 Direct Union of Chord

The *direct union of k copies of Chord* is the structure

$$u_k\text{-Chord} = (\Omega \times \{1, \dots, k\}, \{H_1, \dots, H_k\}, H) ,$$

where H_1, \dots, H_k, H are independent hash functions. Each new node with an identifier Id uses the value $H_i(Id)$ to calculate its position in the i -th copy $C_i = \Omega \times \{i\}$ of the Chord and each document doc is placed at point $H(doc)$ in each copy C_i of the Chord. Notice that we use the same function H to place new documents in distinct copies of the Chord.

Let us consider for a while the structure u_2 -Chord. Suppose that one node leaves a system in an unexpected way, i.e. without sending its resources to its predecessor. When the number of nodes is large, then with high probability both areas controlled by the leaving node in two copies of Chord are disjoint, so no information item stored in the system is lost. Therefore there exists a possibility of restoring lost information items in the first copy of the Chord from the second copy. A similar remark holds for a structure u_k -Chord, where $k \geq 2$. We assume that the structure u_k -Chord is equipped with the procedure for retrieving partially lost information. It should be used by a node when it loses the connection to its immediate successor in each of its copies of the virtual Chord's space.

Let $A = \{n_1, \dots, n_k\} \subseteq \{1, \dots, n + 1\}$ be a random subset of nodes from the structure u_2 -Chord with $n + 1$ nodes. We denote by $K_{A,1}$ the unions of intervals controlled by nodes from A in the first circle and we denote by $K_{A,2}$ the unions of intervals controlled by nodes from A in the second circle. We say that the set A is **safe** if $K_{A,1} \cap K_{A,2} = \emptyset$. Notice that if the set A is safe then no information disappears from the system after simultaneous unexpected

departure of all nodes from A . Moreover, a proper mechanism may be used to recover vanished information from the first circle by information stored in the second circle and conversely. We say that the set A is **unsafe** if A is not safe.

Theorem 3. *Let n be the number of nodes in the structure u_2 -Chord and let A be a set of nodes.*

1. *If $|A| \leq \sqrt{\frac{n}{\ln(n)}}$ then A is safe w.h.p.*
2. *If $|A| \geq \sqrt{n \ln n}$ then A is unsafe w.h.p.*

Notice that the bounds in Theorem 3 does not depend on the number of documents put into the system.

2.3 Folded Chord

Let $\text{succ}(X)$ be the successor of a node in Chord. In the classical Chord protocol each node controls the subinterval $[X, \text{succ}(X))$ of the space Ω . The k -folded Chord, denoted as f_k -Chord, is the modification of the Chord protocol in which each node controls the interval $[X, \text{succ}^k(X))$, where $\text{succ}^1(X) = \text{succ}(X)$ and $\text{succ}^{k+1}(X) = \text{succ}(\text{succ}^k(X))$.

Notice that f_1 -Chord is the classical Chord. Let us consider for a while the structure f_2 -Chord and suppose that a number of nodes in the system is large. Suppose that the small group B of nodes leaves the system in an unexpected way. Then the probability of the event $(\exists b \in B)(\text{succ}(b) \in B)$ is negligible. Consider such a node y that $\text{succ}(y) \in B$. Let x be its predecessor and let $z = \text{succ}(b)$ and $u = \text{succ}(y)$. Note that when the node b leaves the system then z is a new successor of y . Then the node y may send a copy of all information items from the interval $[b, z)$ and may ask the node z for all information items from the interval $[z, u)$. This way we can rebuild the original structure and no information item will be lost. Of course, we can do it if the leaving group B satisfies the property $(\forall b \in B)(\text{succ}(b) \notin B)$. A similar procedure may be applied for f_k -Chord for every $k \geq 2$.

Let $A = \{n_1, \dots, n_k\} \subseteq \{1, \dots, n\}$ be a random subset of nodes from the structure f_d -Chord with n nodes. We say that the set A is **safe** if a distance of every node from A to the next point from A (in the fixed orientation of Chord) is at least d . So, if the set A is safe then no information disappears from the system after simultaneous unexpected departure of all nodes from A and a proper mechanism may be used to recover vanished information from the system. We say, as before, that the set A is **unsafe** if A is not safe.

Theorem 4. *Let n be the number of nodes in the structure f_2 -Chord and let A be a subset of nodes.*

1. *If $|A| \leq \sqrt{\frac{n}{\ln(n)}}$ then A is safe w.h.p.*
2. *If $|A| \geq \sqrt{n \ln n}$ then A is unsafe w.h.p.*

3 The Repairing Process

We shall discuss in this section the dynamics of repairing (complementing) partially removed information from the system. First we introduce the following notation:

- μ - the average number of departures from the system in a second
- T - the average time a node spends in a system (during one session)
- N - the average number of nodes in the system
- u - the proportion of unexpected departures among all departures
- T_r - the average time need for recovery of partially lost information
- N_r - the medium number of nodes waiting for complementing
- δ - the average numbers of documents in the system per one node

Using the Little’s Law form Queueing Theory (see e.g. [5]) we get the following two relations

$$N = \mu \cdot T , \quad N_r = u\mu T_r , \tag{1}$$

so $N_r = uN \frac{T_r}{T}$.

3.1 Safety Bounds

For each of the modifications of the Chord protocol discussed in this paper we found in Section 2 a safety bound S_N . For the r_2 -Chord we have $S_N = \frac{1}{\ln N} \sqrt{\frac{N}{\delta}}$, for u_2 -Chord and f_2 -Chord we have $S_N = \sqrt{\frac{N}{\ln N}}$. If we want to keep our system in a safe configuration with high probabilities, then the following inequality must be satisfied:

$$T_r \leq \frac{TS_N}{uN} . \tag{2}$$

Suppose that $N = 10^5$, $T = 30$ minutes and $u = 0.1$. From Equations 1 we get $\mu = 10^5/1800 \approx 55.55$, so approximately 5.55 nodes unexpectedly leave the system in one second. We shall use this set of parameters to illustrate how we can calculate other parameters of considered systems.

3.2 r_2 -Chord

Suppose that each server stores in its local database the list with items (*doc*, *NoC*, *Info*) for each information item it is responsible. The field *doc* is the description of a document which allows to calculate positions of documents in the Chord structure, *NoC* is the number of copy (equals 0 or 1 in r_2 -Chord) and *Info* is additional information about the document, which usually contains the URL of a node with the source of the document.

We assume that each node periodically sends the command Update(*doc*, 1-*NoC*, *Info*) for each triples (*doc*, *NoC*, *Info*) stored in its local database. The Update command is a simple modification of the original AddDoc command with one additional flag, marking that this is not the AddDoc command, but only the gossiping of stored information.

Let us recall that the classical Chord (see [1]) checks its neighbors periodically with the period of 2 seconds. Therefore we must assume that the first 2 seconds are wasted - during this time the node does not know that its neighbor has left the system. Therefore, there are only $T_r - 2$ seconds for information recovery. Moreover, during this time each node must send approximately δ Update messages.

The following table contains the upper bounds for the time $T_r - 2$ with fixed parameters $N = 10^5$, $T = 30$ min, $u = 0.1$, for different parameters δ :

δ	1	2	3	4	5	6	7
$T_r - 2$	2.9	1.5	0.9	0.5	0.2	0.018	< 0

We see that we are unable to stay in a safe configuration when $\delta \geq 7$. Moreover, if $\delta = 2$ then each node should send all Update messages with frequency of 1.5 seconds, which is smaller than the standard time of checking neighbors - hence we see that the Update messages dominates the whole message traffic in the system. Hence, we see that the r_2 -Chord can stay in the safe configuration only when the total number of documents stored in the system is small.

3.3 u_2 -Chord and f_2 -Chord

In these cases the upper bound for the time T_r is $\frac{T}{u} \sqrt{\frac{1}{n \ln n}}$. Using the same parameters as above we get 16.7757 seconds, so subtracting 2 seconds we obtain 14.7 for the repairing process.

Let us consider the structure u_2 -Chord. During this time the node a should send a request to each node from the second copy of the Chord which can contain information from the gap controlled now by a . The following Theorem shows that there are few such nodes:

Theorem 5. *Let X_1, \dots, X_n and Y_1, \dots, Y_{n+1} be independent and uniformly distributed random variables in the interval $[0, 1]$. Let $L_n = |\{i : Y_i < X_{1:n}\}|$. Then $\mathbf{E}[L_n] = 1$.*

Therefore, we see that the expected number of intervals from the second copy of Chord which have nonempty intersection with a given interval from the first copy is precisely 2 - hence the expected number of nodes which should be asked for its copy of information is 2. Hence the node which wants to recover partially lost information must

1. send few messages to the second copy; the time required for localization of nodes in the second copy equals approximately $\frac{1}{2} \log_2 n \cdot 0.2$ sec;
2. wait for receiving necessary information; the time required for fulfilling this operation equals approximately $2 \cdot \delta \cdot d_s / t_s$ sec where d_s is the size of the information item in the system and t_s is the transmission speed.

So the total repairing time is $T_c = \frac{1}{10} \log_2 n + \frac{2\delta \cdot d_s}{t_s}$. If we assume that each information item stored in the system is of size 0.5 kB and assuming that the

transmission speed $t_s = 500$ kB/s then even if $\delta = 1000$, we need about 5.33 seconds, so we have enough time for finishing this operation within the upper bound, i.e. within 14.7 seconds.

In the structure f_2 –Chord the procedure of localization of copies of documents is more simple: the given node a should ask its actual successor for a portion of information and send some portion of information to its predecessor. The rest of analysis is the same. Obviously, in the f_2 –Chord structure each node X should remember fingers to $\text{succ}^1(X)$ and $\text{succ}^2(X)$ to make the recovery process as short as possible. Using the same parameters as for the u_2 –Chord we calculate that the time needed for finishing the repairing process is about 2.2 seconds.

3.4 Information Lifetime

It is clear that keeping a system in a stable configuration is not sufficient to ensure that the process of recovery will finish successfully. We assume that considered systems were evolving sufficiently long which allows us to use asymptotic results from the renewal theory. This assumption is usually satisfied in practice since P2P protocols evolve for hundreds of days while the average lifetime of nodes is small relatively to the age of system.

A random variable X has the shifted Pareto distribution with parameters α and β ($X \sim \mathbf{Pa}(\alpha, \beta)$) if $\Pr[X > x] = (\frac{\beta}{\beta+x})^\alpha$. If $X \sim \mathbf{Pa}(\alpha, \beta)$ and $\alpha > 1$ then $\mathbf{E}[X] = \frac{\beta}{\alpha-1}$. Notice that an expected value of X is finite if and only if $\alpha > 1$. The variance of the the variable X is finite if and only if $\alpha > 2$ and $\mathbf{var}[X] = \frac{\alpha\beta^2}{(\alpha-2)(\alpha-1)^2}$ in this case.

It was observed by many authors that the distribution of lifetime L of nodes in a real-world peer-to-peer system (i.e. session duration) has a Pareto distribution (see e.g. [6]) with the slope parameter $\alpha \approx 2.06$. It is worth noticing that first investigators report that the value $\alpha \approx 1.06$ but later it occurs that they made a mistake in the interpretation of observed data (see [6] for details).

We model the lifetime as a random variable L with shifted Pareto distribution. Since $\mathbf{E}[L] = \frac{\beta}{\alpha-1}$ and $\mathbf{E}[L] = T$, we get $\beta = (\alpha - 1) \cdot T$. Let us remark that Liben-Nowell et al. in [2] considered a Poisson model of arrivals and departures; the Pareto distribution for modeling nodes' lifetime in a system was used by Derek et al. in [6]. Therefore we will assume that $\Pr[L > x] = \text{pd}(x, \alpha, T)$, where

$$\text{pd}(x, \alpha, T) = \frac{1}{(1 + \frac{x}{(\alpha-1)T})^{\alpha-1}} .$$

The residual lifetime is the conditional random variable of the lifetime conditioned by the event that the modeled object is alive. The cumulative distribution function of the residual lifetime of a random variable L with cumulative distribution F is (see [6])

$$F_R(x) = \frac{1}{\mathbf{E}[L]} \int_0^x (1 - F(t))dt .$$

From this equation we easily deduce that if $X \sim \mathbf{Pa}(\alpha, \beta)$ and X^* is the residual lifetime of the variable X then $X^* \sim \mathbf{Pa}(\alpha - 1, \beta)$. Therefore, if $\beta = (\alpha - 1)T$

and $x \ll T$ then the probability that a given (alive) node will stay in the system for a time longer than x is $(1 + \frac{x}{(\alpha-1) \cdot T})^{-\alpha+1} \approx 1 - \frac{x}{T}$. However, if $\alpha = 2.06$ then the variance of the residual lifetime of a node in a system is infinite.

Let us look at the system from the point of view of one information item put into the system. Therefore let us fix an information item ξ put into the system and let T_ξ be the time of survival of this item in the system.

Theorem 6. *Let us consider the structure u_2 -Chord or f_2 -Chord. Suppose that the parameters of the system guarantee that the system is in a stable configuration for a long period of time. Let T_u be the time necessary to repair all damages after an unexpected departure and let T_n be the time necessary to repair all damages after an expected departure. Then*

$$E[T_\xi] \geq \frac{T}{u(1 - f(T_u)^2) + (1 - u)(1 - pd(T_n)^2)} ,$$

where $f(x) = \frac{T}{T+T_u}pd(x, \alpha - 1, T)$.

4 Applications

Let us consider a medium size structure with $N = 10^4$ nodes. Let us assume that the average lifetime of a node in this system is $T = 30$ minutes. Suppose that the system is in the stable configuration. Then, the parameter $\mu \approx 5.55$, so we see that approximately 5.5 nodes leave and other 5.5 nodes join the system in a second.

Suppose that the parameter $u = 0.1$. This means that approximately 0.55 nodes unexpectedly leave the system in one second. In the classical Chord a given information item can survive no dangerous moment. Dangeorus moments happens every T/u minuts on the average, so the average lifetime of given information item in the classical Chord equals about 300 minutes, i.e. 5 **hours**.

In the system f_2 -Chord we have $T_r = \frac{T}{u} \sqrt{\frac{1}{n \ln n}} \approx 59.3$ sec., so the system is able to stay in a safe configuration for a long time. Let us assume that $T_u = 3.5$ seconds and $T_n = 1.5$ sec. From Theorem 6 we get $E[T_\xi] \geq 87.0965$ hours. Hence, we increase the information lifetime in the system from 5 hours into at least 87 hours i.e. 17 times. This conclusion can be used in a practical way: an owner of an information item who wish this information to remain for a long time in the system should periodically (once five days, in our case) pull this information into the system.

Numerical simulations confirm our theoretical estimations. They show that the average information lifetime in the structure f_2 -Chord (with the parameters as above) is about 100.28 hours but the standard deviation of the information lifetime, as one may expect, is very high.

5 Conclusions

The classical Chord protocol has an unavoidable failure - after an unexpected departure of a node from this system some gathered information may flow out.

Even if we use the technology of duplication of documents gathered in the system then after unexpected departures of $\Gamma(\frac{3}{2})\sqrt{\frac{n}{\delta}}$ nodes (where n is a number of nodes and the total number of documents is δn) some information will be lost with a high probability.

We discussed and investigated three simple modifications of the classical Chord protocol: Chord with duplicated documents, a direct unions of Chord and a folded Chord. Our result suggests that the best way of improving Chord is to use the structure f_k -Chord for small value of parameter $k \geq 2$.

Our last two modifications has also one interesting property: the measure of concentration of length of intervals is much better concentrated than in the classical Chord and this property may improve the effectiveness of some services (like DNS) based on the Chord protocol.

References

1. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: SIGCOMM 2001, San Diego, California, USA (2001)
2. Liben-Nowell, D., Balakrishnan, H., Karger, D.: Analysis of the evolution of peer-to-peer systems. In: ACM Conference on Principles of Distributed Computing, Monterey, CA (2002)
3. Kaiping, X., Peilin, H., Jinsheng, L.: Fs-chord: A new p2p model with fractional steps joining. In: AICT-ICIW 2006: Proceedings of the Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services, Washington, DC, USA, p. 98. IEEE Computer Society, Los Alamitos (2006)
4. Gopalakrishnan, V., Silaghi, B., Bhattacharjee, B., Keleher, P.: Adaptive replication in peer-to-peer systems. In: The 24th International Conference on Distributed Computing Systems (2004)
5. Cooper, R.B.: Introduction To Queueing Theory, 2nd edn. Elsevier North Holland, Inc. (1981)
6. Derek, L., Zhong, Y., Vivek, R., Loguinov, D.: On lifetime-based node failure and stochastic resilience of decentralized peer-to-peer networks. In: SIGMETRICS 2005: Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, pp. 26–37. ACM Press, New York (2007)

A Performance Evaluation of *g*-Bound with a Consistency Protocol Supporting Multiple Isolation Levels*

R. Salinas¹, F.D. Muñoz-Escobedo¹, J.E. Armendáriz-Iñigo²,
and J.R. González de Mendivil²

¹ Instituto Tecnológico de Informática, 46022 Valencia, Spain,
{rsalinas,fmunoz}@iti.upv.es

² Universidad Pública de Navarra, 31006 Pamplona, Spain
{enrique.armendariz,mendivil}@unavarra.es

Abstract. Whereas the Strict Snapshot Isolation (SSI) level is nowadays offered by most of the centralized DBMSs, replicated databases do not usually provide it, since the traditionally considered approach, the pessimistic management, introduces enormous performance penalties that preclude it from production applications. Instead, distributed databases usually offer Generalized Snapshot Isolation (GSI), a more relaxed level, in which a transaction may get a snapshot older than the one that was applied on the database by the time the transaction started.

This paper takes advantage of our MADIS middleware and one of its implemented Snapshot Isolation protocols (SIRC) to design, implement and evaluate the performance of an extended version of SIRC (called *g*B-SIRC). This protocol is able to concurrently execute Generalized Read Committed (GRC), GSI, *g*-Bound —a non-standard SI level limiting the outdatedness of transactions wanting to commit— and optimistic SSI transactions on top of a cluster of centralized DBMSs offering RC and SSI. This work is the first implementation and evaluation of an optimistic SSI level. Although the abort rate of *g*-Bounded transactions is significantly higher than the GSI ones, the performance results show that introducing transactions at more restrictive levels is not detrimental to the completion time or to the abort rate of the transactions using GSI.

1 Introduction

The use of different isolation levels for transactions executing in a database is motivated by the different requirements transactional applications demand on data consistency. This fact yields to a significant performance improvement — less blocking intervals and lower abortion rates— when this is compared with the usage of a single stricter isolation level, e.g. serializable. Some benchmarks,

* This work has been partially supported by the Spanish MEC and EU FEDER grants TIN2006-14738-C02 and BES-2004-6500 and IMPIVA and EU FEDER under grant IMIDIC/2007/68.

as TPC-C [1], encourage the usage of different isolation levels for transactions. Unfortunately, when database replication protocols are designed, it is a common practice to do that only for a single isolation level avoiding the practical impact on flexibility that multiple isolation levels render to applications. There are very few exceptions to this rule [2,3,4]. However, if a database replication system supports more than one level, applications will be able to select the most appropriate isolation level for each transaction, the way it can be done in centralized systems.

Snapshot isolation (SI) [5] is an isolation level based on multiversioning, informally defined as follows. When a transaction starts, a snapshot of the database is taken. Every operation framed inside that transaction can only see the data that was available by the time the snapshot was taken, regardless of the changes that other transactions may have performed simultaneously. When commit is requested, checks take place to ensure that the commit-requesting transaction does not intend to commit changes to any item that was modified in a different transaction during the lifetime of the former (the so-called “first-committer-wins” rule). SI has some valuable properties. First, read-only transactions (often the most numerous ones) are never delayed or aborted. Although SI allows non-serializable histories, most of the serialization anomalies are prevented, and thorough studies exist indicating how to avoid the possible ones [6]. Moreover, its simplicity makes it relatively easy to implement in servers and to be understood by programmers. For this reason, SI has become quite popular in the last years and is offered by most of the vendors.

However, when it comes to extending SI to replicated databases using lazy replication [7], the basic requirement of SSI by which each transaction sees the effects of the transactions that were committed before it —trivial to achieve in centralized databases— becomes an issue, since the replicas are asynchronously updated, and, without a special effort, a client could start a transaction at a replica that has still some pending unapplied writesets. If those changes had been performed by the same client, it would not be able to observe in the new transaction the latest state of the database *just committed by itself*, thus breaking the intended virtual single-database image.

It is important to point out that this phenomenon cannot be observed by a client executing transactions in a sequential order on one or more connections to the same replica, since by the time a transaction has finished, the client is guaranteed to have got its changes applied at the delegate replica, which is called Session Strong SI [8]. The problem arises in clients using stateless intermediaries that do not keep information on the session with the underlying database during the whole client session. Such examples can be found in applications based on HTTP, a stateless protocol. Aiming at load balance, the server might have a pool of database connections served by different replicas. A client could perform an update request, to immediately later perform another request that would end up being run at a different delegate replica that had not committed the previous changes yet. As a consequence, the client would not see the changes that it had just made.

In [8] this problem was considered and new definitions were introduced. Strict SI (SSI) requires transactions to observe the latest snapshot of the database, while Generalized SI (GSI) accepts transactions to observe an older version. In the same way, we can extend the notion of the Read Committed (RC) isolation level to Generalized RC (GRC, where older snapshots can be observed) and Strict RC (SRC, forced to see the latest one) respectively. The trivial approach to supporting SSI implies delaying new transactions until the latest known snapshot has been applied at the delegate replica; this fact gets worse when SRC is considered. The associated overhead dissuades production systems from implementing such protocols.

In this paper, we have extended the certification-based SIRC [3] protocol – initially supporting GSI [5] and GRC levels– (g B-SIRC), to support an optimistic SSI, and a g -Bound [2] series of levels that are intermediate between SSI ($g = 0$) and GSI ($g = \infty$). The g B-SIRC protocol executes a transaction entirely at a given replica (its delegate) and when the transaction requests for its commitment, the writeset (i.e. updates performed by the transaction) is total order broadcast [9] to all available replicas. Upon its delivery at each replica a symmetrical test (certification) against previously certified transactions is run with no communication at all among replicas. This kind of protocol presents the best performance [10], among currently available replication protocols, and the highest degree of decentralization, since each replica runs a symmetrical conflict evaluation phase per delivered transaction. Actually, this is a Read One Write All Available (ROWAA) approach [7], eager-update-everywhere with a single message interaction per transaction.

The aims of this paper are: (a) to provide an optimistic non-blocking implementation of the SSI level in a multilevel consistency protocol offering in replicated environments two isolation levels that are most commonly offered in production centralized databases, (b) to prove that supporting multiple isolation levels does not introduce a significant overhead on transactions' completion time, and (c) to extend the main conclusions of [3], i.e., that relaxed isolation levels reduce abortion rates, to protocols supporting more than two levels. This leads to replication deployments that are able to be as flexible as their centralized counterparts. We have implemented all this support in our MADIS [11] middleware using PostgreSQL as its underlying DBMS. This paper is structured as follows: the system model being used in this paper is described in Section 2. In Section 3 the g B-SIRC protocol is presented. For building it, we took our SIRC with support for GSI and GRC as the basis. An analysis of its performance in MADIS, i.e. the transaction response time and abortion rate, is shown in Section 5. Finally, conclusions end the paper.

2 System Model

We assume a fully replicated system composed of N replicas (R_1, \dots, R_N) where each replica has an underlying DBMS that stores a full physical copy of the database and in order to keep copies consistent a g B-SIRC protocol instance is

also executed. This DBMS is a multiversion one, i.e. a new database version is generated each time a transaction is committed. We assume that the version number is stored in a local replica variable called `last_committed_tid`. The DBMS can concurrently support the execution of transactions under the RC and SI isolation levels.

On the other hand, the SIRC protocol [3] works as follows: upon the start of a transaction T_i , it is tagged with the current `last_committed_tid` at its delegate replica ($T_i.start = last_committed_tid$) and its isolation level ($T_i.si$) and T_i is entirely executed at this replica. At commit time, the interaction with the rest of replicas takes place by propagating the writeset to them using the total order broadcast (read-only transactions are committed with no interaction at all). Upon its delivery at each replica, the writeset is committed if it is successfully certified. The certification process of SI writesets consists in detecting an update conflict between the writeset and the set of previously certified transactions that were concurrent to them (i.e. those transactions whose $T_j.end > T_i.start$). If a conflict arises, the transaction will be discarded (and respectively aborted at its delegate replica) and otherwise sequentially applied and committed (respectively committed at its delegate replica). On the other hand, a delivered RC writeset will be directly validated since the total order delivery avoids dirty writes [12]. However, this successfully certified writeset may conflict with transactions being executed at that replica (those that are still in their local phase) and must be aborted too; several mechanisms have been considered for this issue in middle-ware architectures [13,14]. RC transactions must be more carefully treated, since once they have broadcast their writesets, such transactions should be committed. To this end, their writeset is applied and committed when it is delivered, even in case of being locally aborted by other concurrent transactions once their writeset was broadcast.

2.1 About the g -Bound Isolation Level

This is an isolation level initially defined in [2] that limits the outdatedness between the snapshot gotten by a transaction at its delegate replica and the latest globally committed in the replicated database. This is due to the fact that the same set of transactions are about to be committed; nevertheless, some replicas may run faster than others and, hence, may install snapshots faster than others. There can be many metrics used, in here we consider the presence of conflicts between a writeset to be committed and the readset of every local transaction. The parameter g measures the amount of tolerated conflicting writesets: every time a new transaction is committed, the presence of conflicts between its writeset and the readset of every local g -Bound transaction is checked (in our implementation the readset is defined at table granularity). If conflicts do exist, the conflict count for that local transaction is increased. If the resulting count exceeds its maximum acceptable g , the transaction is aborted. Note that a zero value for g means that such transaction is requesting an SSI level, whilst an infinite g value is like the GSI level defined in [5]. Since this new restriction refers to the objects declared to be *read*, read-only transactions, that would never be aborted

in GSI or under an SSI with pessimistic management, can in g -Bound be indeed aborted. This is a metric example of outdatedness for g -Bound. Other metrics can be the number of data items, tables accessed and many others included in [2] whose overhead and feasibility has to be experimentally tested. This feature has been also studied in [15,16] just to search for a snapshot given its number or its associated timestamp instead of this optimistic approach. The reason for defining this isolation level is that different applications may have different freshness requirements: one application must have an up-to-date query result; another one prefers a low response time but does not care if the reviews are a bit stale; another one does not care about if the result is stale but it requires the entire result to be snapshot consistent, i.e., reflect a state of the database at a certain point of time; or, another application can be satisfied with a weaker version of this guarantee, requiring only that information retrieved about a data item reflects the same snapshot whereas different data items can be from different consistent snapshots.

<pre> Initialization: 1. lastvalidated_tid := 0; 2. lastcommitted_tid := 0; 3. ws_list := \emptyset; 4. tocommit_queue := \emptyset I. Upon operation request for T_i from local client 1. if select, update, insert, delete a. if first operation of T_i /* T_i includes $\langle g, tables_read \rangle$ */ - $T_i.conflicts := 0$ - $T_i.decision := commit$ - $T_i.RS := \emptyset$ - $T_i.aborted := FALSE$ - $T_i.si := FALSE$ - $T_i.start := lastcommitted_tid$ - multicast $T_i.ID$ in total order b. if $T_i.aborted = FALSE$ - execute operation at R_n c. return to client 2. else /* commit */ a. if $T_i.aborted = FALSE$ - $T_i.WS := getWriteset(T_i)$ from local R_n - if $T_i.WS = \emptyset$, then commit and return - multicast T_i using total order II. Upon receiving $T_i.ID$ 1. if T_i is local in R_n a. append $T_i.ID$ to tocommit_queue 2. else discard message </pre>	<pre> III. Upon receiving T_i 1. if $T_i.g \neq RC \wedge \exists T_j \in ws_list : T_i.start < T_j.end$ $\wedge T_i.WS \cap T_j.WS \neq \emptyset$, then a. if T_i is local then abort T_i at R_n else discard 2. else a. $T_i.end := ++lastvalidated_tid$ b. append T_i to ws_list and tocommit_queue IV. Upon $T_i == head(tocommit_queue)$ 1. remove T_i from tocommit_queue 2. if T_i is a $T_i.ID$ message a. $T_i.si := TRUE$ b. if $T_i.aborted = TRUE$ - restart T_i /* All its operations must be restarted, including step I.1.a. */ c. return 3. else a. $\forall T_j : T_j$ is local in $R_n \wedge T_j.si = FALSE$ $\wedge T_j.aborted = FALSE$: - $T_j.conflicts := T_j.conflicts +$ $getConflicts(T_i.WS, T_j.tables_read)$ - $T_j.aborted := (T_j.conflicts > T_j.k)$ b. if T_i is remote at R_n * begin T_i at R_n * apply $T_i.WS$ to R_n * $\forall T_j : T_j$ is local in $R_n \wedge T_j.WS \cap T_i.WS \neq \emptyset$ $\wedge T_j$ has not arrived to step III - abort T_j c. commit T_i at R_n d. $++lastcommitted_tid$ </pre>
---	--

Fig. 1. g B-SIRC algorithm at replica R_n

3 Protocol Description

In addition to the isolation levels provided by SIRC [3] (GRC and GSI), the g B-SIRC protocol supports g -Bound SI with different degrees of optimistic outdatedness limitation, being $g = 0$ equivalent to the SSI. As already noted, we have followed the algorithm presented in [2] to perform an extension of SIRC and

derive the g B-SIRC protocol. Its informal description showing the most important events that happen at each replica can be seen in Figure II; these events will be used to describe how the protocol works in a more detailed manner. Before the actual execution of the first instruction of every g -Bound transaction T_i , a $T_i.ID$ message is total order broadcast (step I.1.a in Figure II) and the data structure associated to the transaction initialized. At this moment, T_i gets the snapshot from its DBMS delegate replica that can be older than the desired one ($T_i = \text{last_committed_tid}$). In any case, the first operation and subsequent operations associated to the transaction are executed (I.1.b). Whenever the T_i requests its commit operation, the writeset is collected and total order broadcast to the rest of replicas to independently decide its outcome (I.2).

Upon the delivery of $T_i.ID$ message at its respective delegate replica (the rest silently discard it), it is enqueued in `tocommit_queue` (II). The delivery of T_i (III) depends on the isolation level for its certification: an SI one (and its associated flavors) performs the first-committer-wins rule (III.1); whereas an RC transaction is certified as soon as it is delivered (see details in III). It is worth noting that $T_i.end$ is set. A successfully certified transaction is stored in `tocommit_queue` waiting to be applied (and committed) and in `ws_list` so that further delivered writesets take into account this newly delivered writeset for their certification. Messages stored in `tocommit_queue` are sequentially treated and can be of two kinds (IV). The first one to be considered is the $T_i.ID$ message, if T_i has already been aborted it is the time to get it restarted. Otherwise, it is a writeset message that has to be applied and committed. Nevertheless, g B-SIRC checks for the outdatedness of local transactions T_j (in terms of tables read by T_j) executed under g -Bound at R_n against its $T_i.WS$ (IV.3.a) provoking the abortion of some of them due to their associated g values. Every g -Bound transaction requesting its commit needs to have had its $T_i.ID$ message (and thus, $T_i.si = \text{TRUE}$). Very short transactions, with a duration shorter than the time required to apply every writeset that was present in the `tocommit_queue` by the time they started, have most likely to wait. Finally, a pessimistic implementation of SSI would have blocked the client's start until this moment. On the contrary, the client is optimistically allowed to start in g B-SIRC, hoping that no conflicts will appear most of the times.

4 Test Conditions

The g B-SIRC protocol has been implemented in our MADIS replication middleware. MADIS is written in Java and offers a standard JDBC interface to its clients, providing them seamlessly with a virtual single database that is actually kept consistently replicated. The required changes to the applications are limited to loading a different JDBC driver and using a different URL. The Spread toolkit was the group communication system of choice and PostgreSQL 8.3.1 (with no modification) as the DBMS. In order to take advantage of the main benefit of a replication middleware, i.e., the decoupling from the underlying database manager, the writeset extraction system as well as other services needed from the

database are entirely performed by means of JDBC primitives. MADIS features a Block Detector [14] that simplifies deadlock resolution and allows a prompter abortion of transactions whose outcome is known in advance. The experiments were run on a 8-node cluster interconnected by a Gigabit-ethernet switch. Each node has a Pentium IV 2.8 GHz processor with 1 GB of main memory.

In order to simplify the seek of a certain conflict rate we used the hotspot approach [17]. In this model, the entire database is split up into two sections: the hot spot and the low-conflicting area. Two parameters define the usage of the hotspot: the fraction of the total number of elements in the database and the fraction of the sentences that will access elements in this area of high concurrency access. Before starting the tests, the database is populated. The total number of items in the database has been set at 10000 rows. At each replica, a client is run, which uses the local server. Each client process launches a variable number of threads that try to satisfy a certain transaction rate (specified later for each experiment). If the queue of pending jobs grows beyond a certain threshold, the system is considered to be overloaded under the load being tested, and the experiment is aborted. The results taken are based upon a stationary regime. The time measurements of the first transactions are discarded, in order to avoid the initial transient. PostgreSQL shows a remarkable worsening in the timings after the first transactions. In order to get minimally stable results, each experiment has run 10000 transactions in total. The test performed was to run a series of *jobs*, each job consisting of the following operations: (a) A certain number of reads on two randomly chosen tables, out of four available in total. (b) A 100 milliseconds wait. (c) A certain number of updates on two randomly chosen tables. After each write, a small wait (100 ms divided by the number of writes) is done. Transactions are not retried when aborted. We kept the block detector poll interval at one second, which yields a reasonable response to deadlocks between the DBMS and the middleware while not overloading the DBMS. In our tests we wanted to analyze the performance of *gB-SIRC* when running transactions at the different isolation levels it supports. The performance is measured by means of the following parameters: (i) *Response time*, the average amount of time needed for succeeding transactions to complete; (ii) *Abortion time*, the average amount of time used by aborted transactions; and, (iii) *Abortion rate*, the ratio between aborted transactions and the total being submitted (0..1).

5 Performance Results

Varying the g Parameter in *gB-SIRC*. The TPS were fixed to 16 just to see the influence of varying the g parameter between 0 (SSI) and 7; results can be seen in Figure 2. We can quite clearly distinguish a threshold value of g ($g_{thres} = 3$), from which GSI and *g*-Bound SI behave equally. This is strongly related to the average length of `tocommit_queue`; this queue tends to increase its size in average. Hence, the distance between the start moment of new transactions, and the actual snapshot they are getting, becomes bigger. In an ideal, though unrealistic, case where local and remote transactions could be applied in a null period of time, the snapshot taken by a transaction T_i would always be the one

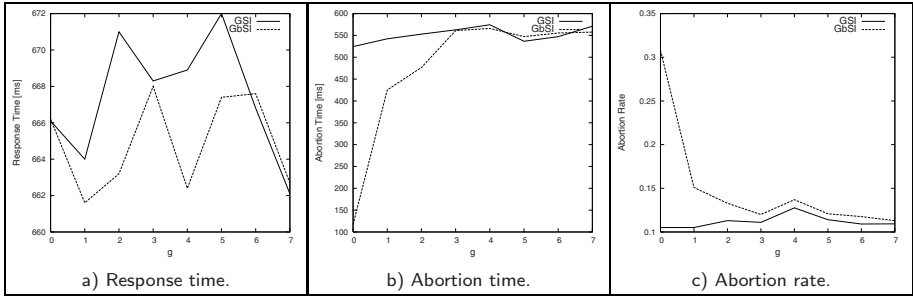


Fig. 2. g B-SIRC with different g values

wanted, and the g -Bounding would make no difference whatsoever, except for the small overhead produced by the $T_i.ID$ message processing. The response time, shown in Figure 2.a, varies very little, between 662 ms and 672 ms, which is a negligible 1.51%, showing no difference between GSI and g -Bound SI. So, transactions that successfully finish do not exhibit timing differences. With regard to the abortion time, see Figure 2.b we observe that the abortion time in the g -Bound level is smaller than that of GSI transactions in the smallest values of g (a 20% of variation between a GSI and a $g = 0$ (SSI) transaction). Note that the abortion of a g -Bound transaction depends on the number of conflicting writesets that have been applied whilst is still running (IV.3.a in Figure 1) (i.e. a single delivered conflicting writeset with SSI) whilst in GSI such abortion is delayed until certification time (III in Figure 1). Note that the MADIS block detector [14] can be tuned for aborting as soon as possible GSI transactions, but we have used it with a long interval in these tests in order to only ensure liveness, leaving abortion decisions to the replication protocol. As soon as g values are increased, the g -Bound abortion time is also increased until it gets values like those of GSI, once the g_{thres} value is reached. Finally, Figure 2.c shows that the abortion rate decreases as g increases. It is worth mentioning that for every $g > g_{thres}$, the abortion rate differences are minor than 2% between GSI and g -Bound levels. Note that in this experiment we have used the same sequence of transactions for both isolation levels for each given g value, but that sequence was also different between different g values.

Combining All Isolation Levels. In this experiment, transactions at every level offered by this protocol were concurrently run, in the same proportion. The transaction rate varied from 8 to 22 TPS and results are presented in Figure 3. In regard to the g -Bound level, we chose the levels SSI ($g = 0$), $g = 1$ and $g = 2$ (recall that $g_{thres} = 3$) as greater g values are equal to the GSI level. If we take a look at the response time (see Figure 3.a), the commit time for such isolation levels. GRC obtains the worst times, but all other levels can be bunched together since no clear winner can be found; i.e. it highly depends on the total order message delivery. Regarding abortion time (Figure 3.b), the best results correspond to the most restrictive isolation level (SSI) and, on the contrary, the worst ones

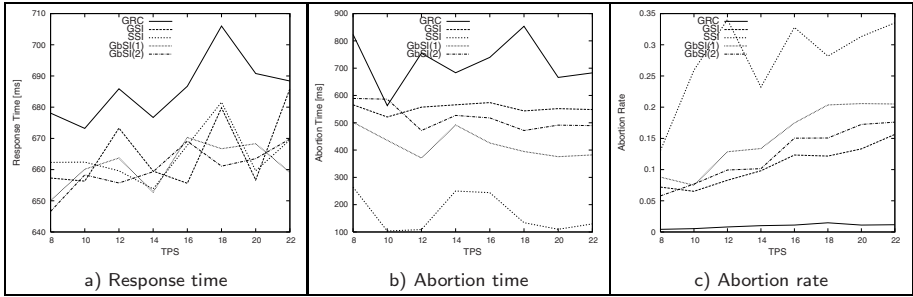


Fig. 3. Combining transactions at all levels

corresponds to GRC. Both previous metrics are due to the certification test associated to each isolation level (see Section 2). Finally, all isolation levels can be ordered in the same way: the most relaxed the isolation level is, the lowest abortion rate can be obtained. This can be inferred from Figure 3.c.

Recapitulation. The snapshot outdatedness limitation given in the g -Bound levels makes the abortion rate much higher, especially in highly loaded systems. Note that the abortion time depends only on the arrival rate for conflicting transactions; in other words, it does not depend on the delivered writeset isolation level as writesets being applied are independently treated by g B-SIRC (step IV.3.a in Figure 1). However, the clue here is how to infer an adequate definition of *conflict* (in our case, based on tables read) for g -Bound. The finer it is the lower abortion rates it produces. Nevertheless, there is a trade-off since the overhead produced by its checking process would make it unfeasible (think about a row-level granularity) and other alternatives must be considered, e.g. stored procedures. On the other hand, read-only g -Bound transactions whose execution time is shorter than their start message processing time will remain blocked until these messages are treated; i.e. it is still needed to verify their g values (their *conflicts* with previous update transactions) so that they are still g -Bounded. This fact, along with a reduction of the response time for update transactions too, can be alleviated if non-conflicting certified writesets are concurrently applied as suggested in [13] and/or by properly tuning the block detector [14].

6 Conclusions

In this paper we have analyzed the performance of the g B-SIRC protocol in a replication middleware that works on top of a standard DBMS. It is an extension of SIRC [3], originally suited for executing GRC and GSI transactions, that supports the g -Bound SI level [2] –which establishes a limit on the outdatedness on the snapshot taken by transactions wanting to commit, ranging from SSI to GSI–. This new protocol offers an optimistic, non-blocking SSI, which is, as far as we are aware, the first implementation of a practical SSI replication protocol in a middleware system. The overall results show that multiple isolation levels can be

supported simultaneously in a single replication protocol without compromising the performance of any of them. Application programmers can select the most appropriate isolation level for each transaction, keeping a lower abortion rate for less-demanding transactions, thus bringing to replicated databases a feature that had been exclusive of centralized databases for a long time. Despite its benefits for applications requiring such high guarantees, care must be taken when using g -Bound levels, since the coarse granularity used (table level) notably increases the abort rate in heavily loaded applications.

References

1. Transaction Processing Performance Council: TPC Benchmark C - standard specification. Version 5.8 (2007), <http://www.tpc.org>
2. Armendáriz, J.E., Juárez, J.R., González de Mendivil, J.R., Decker, H., Muñoz, F.D.: k -Bound GSI: A flexible database replication protocol. In: SAC, ACM, New York (2007)
3. Salinas, R., Bernabé, J., Muñoz, F.: SIRC, a multiple isolation level protocol for middleware-based data replication. In: ISCIS, IEEE Computer Society Press, Los Alamitos (2007)
4. Juárez, J.R., Armendáriz, J.E., González de Mendivil, J.R., Muñoz, F.D., Garitagoitia, J.R.: A weak voting database replication protocol providing different isolation levels. In: NOTERE, pp. 159–171 (2007)
5. Elnikety, S., Pedone, F., Zwaenepoel, W.: Database replication providing generalized snapshot isolation. In: SRDS (2005)
6. Fekete, A., Liarokapis, D., O’Neil, E., O’Neil, P., Shasha, D.: Making snapshot isolation serializable. *ACM TODS* 30(2), 492–528 (2005)
7. Gray, J., Helland, P., O’Neil, P.E., Shasha, D.: The dangers of replication and a solution. In: SIGMOD (1996)
8. Daudjee, K., Salem, K.: Lazy database replication with snapshot isolation. In: VLDB (2006)
9. Chockler, G., Keidar, I., Vitenberg, R.: Group communication specifications: a comprehensive study. *ACM Comput. Surv.* 33(4), 427–469 (2001)
10. Wiesmann, M., Schiper, A.: Comparison of database replication techniques based on total order broadcast. *IEEE TKDE* 17(4), 551–566 (2005)
11. Irún, L., Decker, H., de Juan, R., Castro, F., Armendáriz, J.E., Muñoz, F.D.: MADIS: a slim middleware for database replication. In: Cunha, J.C., Medeiros, P.D. (eds.) Euro-Par 2005. LNCS, vol. 3648, Springer, Heidelberg (2005)
12. Berenson, H., Bernstein, P., Gray, J., Melton, J., O’Neil, E., O’Neil, P.: A critique of ANSI SQL isolation levels. In: SIGMOD (1995)
13. Lin, Y., Kemme, B., Patiño-Martínez, M., Jiménez-Peris, R.: Middleware-based data replication providing snapshot isolation. In: SIGMOD (2005)
14. Muñoz, F.D., Pla, J., Ruiz, M.I., Irún, L., Decker, H., Armendáriz, J.E., González de Mendivil, J.R.: Managing transaction conflicts in middleware-based database replication architectures. In: SRDS (2006)
15. Plattner, C., Wapf, A., Alonso, G.: Searching in time. In: SIGMOD, pp. 754–756 (2006)
16. Guo, H.: “Good Enough” Database Caching. PhD thesis, U. of Wisc (2005)
17. Kemme, B.: Database Replication for Clusters of Workstations. PhD thesis, ETHZ (2000)

Integrity Dangers in Certification-Based Replication Protocols*

M.I. Ruiz-Fuertes¹, F.D. Muñoz-Escó¹, H. Decker¹, J.E. Armendáriz-Iñigo²,
and J.R. González de Mendivil²

¹ Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
Camino de Vera, s/n
46022 Valencia, Spain

{miruifue, fmunyoz, hendrik}@iti.upv.es

² Depto. de Ing. Matemática e Informática
Universidad Pública de Navarra
Campus de Arrosadía, s/n
31006 Pamplona, Spain

{enrique.armendariz, mendivil}@unavarra.es

Abstract. Database replication protocols check read-write and/or write-write conflicts. If there are none, protocols propagate transactions to the database, assuming they will eventually commit. But commitment may fail due to integrity constraints violations. Also, the read actions of integrity checking may give raise to new conflicts. Thus, some more care must be taken if, in addition to the consistency of transactions and replicas, also the consistency of integrity constraints is to be maintained. In this paper, we investigate how certification-based replication protocols can be adapted to correctly and transparently deal with the built-in integrity support provided by the underlying DBMS. Also, we experimentally demonstrate the negative effects that an incorrect management of integrity constraints causes in a database replication distributed system.

1 Introduction

Many database replication protocols have been proposed [1,2,3] over the years. None of them has assessed the support of semantic consistency as postulated by integrity constraints. In general, ignoring problems related to integrity checking in concurrent, distributed and replicated systems is in good company. Even the most well-known authors in the field of transaction processing are accustomed to assume that all transactions are programmed in such a way that they preserve integrity when executed in isolation, and therefore, integrity preservation is also guaranteed by serializable schedules of concurrent executions [4,1].

Unfortunately, this assumption does not always apply. And even if it would, its consequence does not necessarily hold in replicated databases. Concurrent

* This work has been partially supported by EU FEDER and the Spanish MEC under grants TIN2006-14738-C02 and BES-2007-17362, and IMPIVA and EU FEDER under grant IMIDIC/2007/68.

transactions may start and execute in different nodes, and proceed without problems until they request commitment. Upon receipt of the commit request of a transaction, the replication protocol validates it and guarantees its commitment if there is no read-write or write-write conflict among concurrent transactions. Complications may arise if constraints are checked in deferred mode, i.e., at effective commit time after conflict validation by the replication protocol. Then, integrity checking may diagnose constraint violations by transactions that are already successfully validated by the protocol, i.e., the protocol has already sanctioned those transactions to commit. Moreover, accesses made by integrity checking remain unnoticed by the protocol.

An example of integrity violation would be the following. Suppose a table constraint like `FOREIGN KEY x REFERENCES t.y`, where `t` is the referenced table and `y` is the primary key column of `t`. If a transaction inserts a row with value `x = v`, and a concurrent transaction in another node deletes the row in `t` with column value `y = v`, a violation of the foreign key occurs when the protocol tries to commit both transactions in the same node.

One could think that a possible solution is to include the accesses made by an immediate integrity checking –i.e. integrity is checked upon each update action– in the transaction’s readset, or even in the writeset as suggested in [5], in order to be considered in the conflict checking phase of the replication protocol. This way, two transactions accessing the same item, either for updating or for integrity checking, will be identified as conflicting and one of them will be aborted by the protocol. It is easy to see that this will increase the abortion rate unnecessarily, as two transactions reading the same object for integrity checking do not necessarily entail an integrity violation –e.g. two transactions inserting rows with a foreign key to the same referenced row–. Besides this, integrity checking is an internal DBMS process and there is no standard nor direct way to know exactly what objects have been accessed during this process. Moreover, immediate integrity checking seems not to be the best option. Indeed, in a centralised setting, integrity constraints have been traditionally managed using a deferred or delayed checking [6,7,8], instead of an immediate one. This can be explained both in terms of performance improvements [6,8] and because such checking admits temporary inconsistencies that can be solved before transactions end [6], avoiding intermediate checkings that could have uselessly required time and resources. Moreover, there are transactions that can be correctly managed with deferred checking but cannot be with immediate checking, because every possible order of their operations leads to an intermediate integrity violation [9].

Note also that the semantics of *immediate checking* are unclear [8]. To preserve serializability, each transaction should use the appropriate long read and long write locks on all items it has read or updated. When another concurrent transaction immediately checks constraints on any of its updates, which kind of isolation level should be used in such check? If serializable is used, none of the concurrent accesses will be observable; if other levels are used, as either *read committed* or *snapshot*, as analysed in [8], the serializable guarantees might be broken. As a result, deferred checking seems to be the safest approach.

The aim of this work is to appropriately extend the current integrity support available in centralised DBMSs to dependable distributed systems that use replicated databases. Among all currently available database replication protocols, certification-based ones [3] have the best performance and they are also the most decentralised ones since they use a symmetrical conflict evaluation phase that can be executed without further communication between replicas (apart from the regular writeset propagation, common to all protocol classes). So, focusing on decentralisation and reliability we explore how such integrity management support available in DBMSs can be extended to middleware-based database replication systems that use certification-based protocols. This type of protocol needs at least two steps for transaction management. A first step devoted to transaction conflict checking and, if such check succeeds, a second step where remote updates are applied in non-delegate replicas and transactions are finally committed. If constraint checking is deferred till transaction commit time, some misbehaviour could be generated: the protocol could have admitted a transaction as correct in such first transaction conflict check –reporting its success to the client application– and later the underlying DBMS might abort it when its commitment is requested. So, deferred checking was the solution to some problems in centralised systems, but it is now the origin of others in decentralised scenarios.

The rest of this paper is structured as follows. Section 2 explains in detail the certification-based replication and the extensions made for a proper management of integrity constraints. In Section 3 we present the experimental results obtained. Section 4 analyses the problems that other works present with regard to integrity consistency. Finally, Section 5 concludes the paper.

2 Certification-Based Database Replication Protocols

Most modern database replication protocols use total order broadcast for propagating sentences or writesets/readsets of transactions to other replicas [3]. Among them, certification-based replication (abbr., CBR) protocols provide good performance by optimised algorithms, such as [10].

Readset collection and propagation can be costly if row-level instead of table-level granularity is used. So, in practice, CBR is rarely used for implementing serializable isolation. On the other hand, CBR is the preferred protocol class when the *snapshot* isolation (abbr., SI) level [11] is supported, mainly because this level relies on multiversion concurrency control, and readsets do not need to be checked during certification. However, since such certification is based on logical timestamps and depends on the length of transactions, a list of previously accepted certified transactions is needed for certifying the incoming ones.

So, we focus on SI-oriented CBR protocols in this paper. As defined in [3], a general protocol of this kind is displayed in Figure 1a, where r_i references the local replica executing the protocol. Initially, a transaction t is locally executed in r_i . When t requests commitment, its writeset is collected and broadcast in total order to the set R of alive replicas, along with the identifier of its delegate replica, r_i . Upon the reception of such message, each replica certifies the incoming

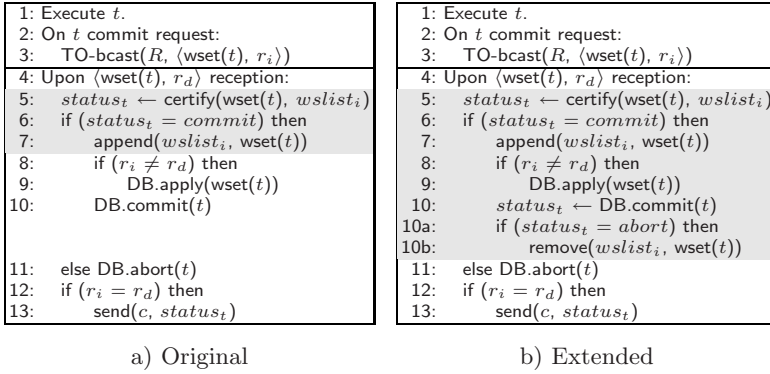


Fig. 1. SI certification-based protocols

writeset against concurrent transactions, looking for write-write conflicts (line 5). This certification process returns a *commit* result if no conflicts are found, or an *abort* result otherwise. Such result is assigned to $status_t$. Note that this validation stage is symmetrical since each replica holds the same history list of previously delivered and successfully certified writesets, $wslis_i$. If no conflict is found, the writeset is included in the history list and the local DBMS interface, DB , is used to apply the writeset in non-delegate replicas and to commit it in each node –If writeset application is impeded, e.g. by t being involved in a deadlock and aborted by the DBMS, it is reattempted until it succeeds–. On the other hand, if a conflict appears during certification, t is aborted in its delegate and discarded in the rest of replicas –for simplicity, the pseudocode represents both actions as the $DB.abort(t)$ operation, although no operation is requested to the DBMS in non-delegate replicas–. Finally, the delegate informs the client c with the transaction outcome, represented by $status_t$.

A basic data structure in CBR protocols is the history list. A writeset should be added to that list in step 7 of the protocol, once it has been accepted for commitment. Thus, the list might grow indefinitely. To avoid that, the list can be pruned following the suggestions given in [3]. Accesses to this list are confined within mutually exclusive zones, shaded in gray in the pseudocode.

As already indicated, certification-based replication gives rise to several problems with regard to integrity constraints. If there is any deferrable declarative constraint and a transaction requests deferred checking, that checking will be done by the DBMS at line 10 right before the actual commit operation. However, as we have seen, that may lead to constraint violations and unexpected abortions. In this case, any repeated attempts to commit the transaction clearly would be in vain. Also note that in line 6 the transaction was assumed successful, and other transactions whose data were delivered after t may have already been aborted due to t 's assumed commitment. So, certification-based protocols need to be modified in order to correctly deal with deferrable constraints.

The extensions for managing integrity constraints in SI CBR protocols, as displayed in Figure 1b, seem to be minor. Only a slight modification of the

original line 10 is needed, for recording the result of the commit attempt, updating the value of $status_t$ in order to represent the real final outcome of each transaction –DB.commit(t) returns a result of *abort* if commitment failed due to integrity violation, or a *commit* result if the operation ended successfully–. This way, although a transaction t successfully passes its certification phase, having its $status_t$ temporarily set to *commit* at line 5, this status changes to *abort* if t violates integrity when tried to be committed in line 10. Moreover, when such commit attempt fails due to integrity violation, the writeset of t is removed from the $wslist_i$, since it has not been finally accepted. This is done in lines 10a and 10b. With these simple and easy-to-implement modifications we are able to completely eliminate a common incorrect behaviour of replication protocols. Moreover, this solution seems to be the only feasible one, since solutions based on increasing the readset or writeset with accesses made by integrity checking present the problems already mentioned.

However, these seemingly minor extensions may have a notable impact on system performance. Typical SI CBR protocols [12,10,13] use some optimisations in order to achieve good performance. One of them consists in minimising the set of operations to be executed in mutual exclusion in the part of the protocol devoted to managing incoming messages (the related protocol section in Figure 1a only encompasses lines 5 to 7). In many protocols (e.g., [10]), an auxiliary list is used for storing the writesets to be committed. As a result, new certifications can be made, once the current writeset has been accepted. With our extensions, no new writeset can be certified until a firm decision on the current one has been taken. That only happens after line 10b in Figure 1b; i.e., once the writeset has been applied in the DBMS and its commitment has been requested. This might take quite some time, and must be done one writeset at a time.

3 Experimental Analysis

For analysing the proposed extensions by practical experiments, we implemented both SI CBR protocol versions in our replication middleware MADIS [14]: a) IntUnaware –integrity-unaware protocol– corresponds to the pseudocode of Figure 1a with only two modifications: first, it is able to identify those transactions that raise integrity exceptions when tried to be committed and so it does not indefinitely reattempt them (thus we obtain a protocol that keeps liveness although it still improperly manages integrity consistency), and second, it informs clients of the real final status of transactions in line 13; and b) IntAware –integrity-aware– protocol, which corresponds to the pseudocode of Figure 1b.

In short, the integrity management error made by the IntUnaware protocol is to keep in the history list those transactions that were aborted due to integrity violations. Although this version is incorrect, we wanted to analyse the difference of performance between existing protocols such as the IntUnaware protocol and our proposal to make it correct, as embodied by the IntAware protocol.

System Model. We assume a partially synchronous distributed system where each node holds a replica of a given database. For local transaction management, each

node has a local DBMS that provides the requested isolation level and supports integrity maintenance by reporting errors in case of constraint violation. On top of the DBMS, a database replication middleware system is deployed. This middleware uses a group communication service (abbr., GCS), that provides a total order multicast.

Test Description. To accomplish the analysis, we use Spread [15] as our GCS and PostgreSQL [16] as the underlying DBMS. Transactions access a database with two tables, *tbl1* and *tbl2*, each with 2,500 rows and two columns. The first column of each table is its primary key. The second column of *tbl1* is an integer field that is subject to updates made by transactions. The second column of *tbl2* is a foreign key, checked in *deferred* mode, that references the first column of *tbl1*. This foreign key constraint defines the integrity consistency of our database.

Two types of transactions are used: a) transactions that preserve integrity, called IntS –integrity safe– transactions; and b) transactions that violate an integrity constraint. These last transactions update the foreign key column of *tbl2* with a value that has no referenced value in the primary key column of *tbl1*, and are called IntV –integrity violating– transactions. In our analysis, we have varied the proportions of IntS and IntV transactions: we analysed test runs with 0, 3, 6, 9, 12 and 15% of IntV transactions –higher percentages would not be realistic and so are not considered–, to measure the consistency degradation as more and more transactions are improperly added to the history list. Thus, all shown graphs display this percentage in their x axis.

Both protocols have been tested using MADIS with 2 replica nodes. Each node has an AMD Athlon(tm) 64 Processor at 2.0 GHz with 2 GB of RAM running Linux Fedora Core 5 with PostgreSQL 8.1.4 and Sun Java 1.5.0. They are interconnected by a 1 Gbit/s Ethernet. In each replica, there are 4 concurrent clients, each of them executing a stream of sequential transactions, with a pause between each pair of consecutive transactions. Two environments have been studied: a low-loaded one, with 100 ms of pause between transactions; and a high-loaded environment, with no pause between transactions. Both IntS and IntV transactions access a fixed number of 20 rows from table *tbl1* for writing –these accesses may cause conflicts between transactions, which will be detected during certification–. Besides this, every transaction also updates a row from table *tbl2*: IntS transactions do it preserving integrity, whilst IntV ones always violate the related foreign key constraint.

Evaluating Incorrect Decisions. In order to clearly show the differences in the decisions made by each protocol, one replica node works with the IntAware version and the other one uses the IntUnaware protocol. So, for convenience, we may identify nodes as aware or unaware node, respectively. With this configuration, it is easy to detect the incorrect decisions made by the IntUnaware protocol. Suppose that an IntV transaction T_v is delivered in the system, presenting no conflicts with concurrent transactions. The aware node tries to commit it and discards it when the database notifies the integrity violation, removing T_v from the history list. In the unaware node, although the integrity violation is detected and so T_v is not indefinitely retried, it is not deleted from the history list.

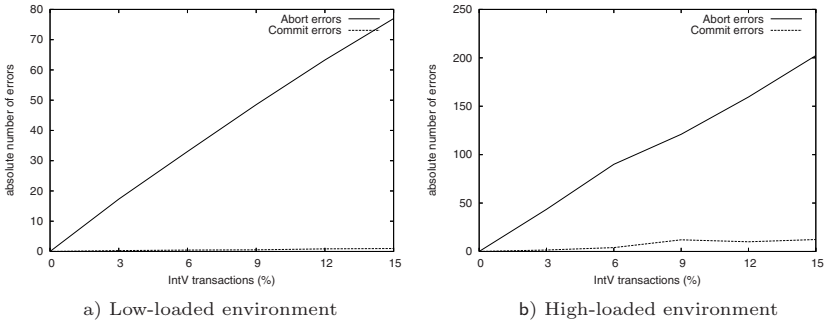


Fig. 2. Errors of the IntUnaware protocol

Problems arise when a subsequent IntS transaction presents write conflicts only with IntV transactions. In the unaware node, this transaction will not successfully pass the certification phase and thus will abort, while it will commit in the aware node. Let us call this an *abort error*. A transaction T_a incorrectly aborted in the unaware node is committed in the aware one. This way, it appears in the history list of the aware node but not in the history list of the unaware one. Now suppose that a subsequent IntS transaction T_c is delivered. If T_c only presents conflicts with T_a , T_c will abort in the aware node but commit in the unaware one. This is a *commit error*. Both abort and commit errors were computed in the tests and are shown in Figure 2 in absolute numbers over the total of transactions issued –16,000 in each test. Notice that only IntS transactions are subjected to such errors, as IntV transactions always end in abortion.

Mainly, detected errors consist in abort errors, i.e. aborting transactions that conflict with others incorrectly included in the history list. Commit errors are less usual as transactions in an unaware node are certified against a greater number of transactions, thus being more likely to get aborted by mistake. The graphs show that, as expected, the greater the percentage of IntV transactions, the greater number of errors made by the unaware node contrary to the always correct behaviour of the aware node.

Evaluating the Length of Transactions. Figure 3 shows the length of local committed transactions (in ms.). Figure 3a, corresponding to a low-loaded environment, shows that no important difference appears between the protocols, as the arrival rate is low enough and transactions do not have to wait for accessing the mutual exclusion zone. On the other hand, when the load becomes higher (Figure 3b), it is clearly seen that the IntAware protocol performs worse than the IntUnaware one. Indeed, completion times increase up to 16.18% in the case of 12% of IntV transactions.

Recall that the proper management of integrity constraints prevents the IntAware protocol from applying any optimisation proposed for certification-based replication protocols. Moreover, it has to be noticed that the IntUnaware protocol only applies the optimisation consisting in certifying newly delivered writesets concurrently with the application of previous ones. Thus, this difference

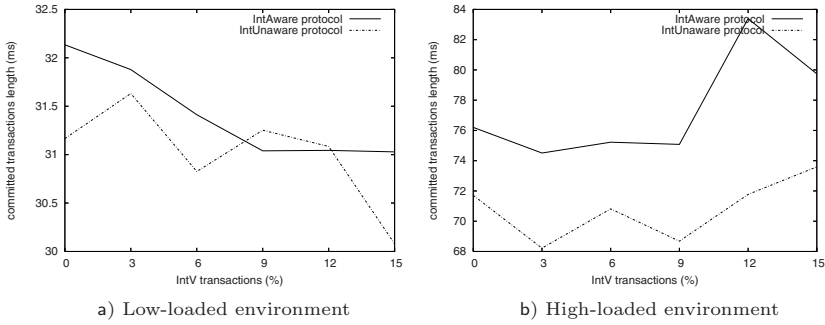


Fig. 3. Length of committed transactions

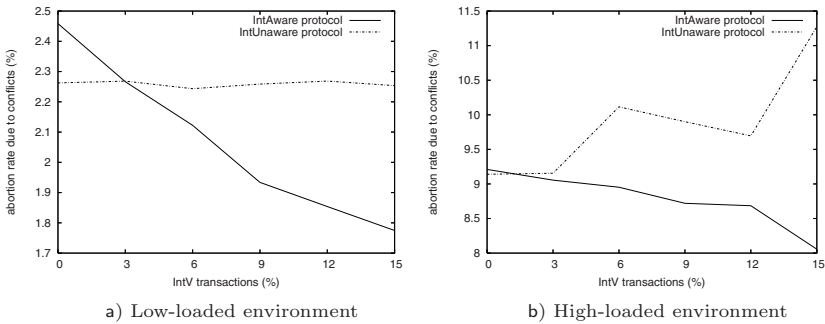


Fig. 4. Abortion rate due to actual certification conflicts

would be bigger when comparing with an optimised version of the SI CBR protocol type. Likewise, when increasing the number of replicas and clients, the performance gap will also increase.

Evaluating the Abortion Rate. Figure 4 shows the average percentage of local transactions that are aborted in a node due to write conflicts with concurrent transactions previously delivered. These conflicts are detected during certification phase and are not directly related to integrity consistency. But, as seen before, the history lists of both protocol versions differ when transactions involve some integrity violation. This may lead to different sets of transactions to be checked against the one being certified. This way, the abortion rate differs from aware to unaware nodes.

In a low-loaded environment, the abortion rate remains constant in the Int Unaware protocol, while it linearly decreases in the IntAware node, in which, as the percentage of IntV transactions increases, more and more transactions are removed from the history list due to integrity violation, leading to a smaller probability for local transactions to present conflicts with the remaining ones. In a high-loaded environment, this behavioural trend is maintained; i.e., the IntAware protocol aborts less transactions than the IntUnaware one, as Figure 4b shows.

4 Related Work

As seen in the previous section, the improper management of integrity-violating transactions leads to a higher abortion rate. This effect will be increased when exploiting an optimisation proposed for certification-based protocols [13], consisting in grouping multiple successfully certified writesets, applying all of them at once in the underlying DBMS. This reduces the number of DBMS and I/O requests, thus improving a lot the overall system performance. On the other hand, it will also generate the abortion of all transactions in a batch as soon as one of them raises an integrity constraint violation.

We have also remarked that indefinitely reattempting to commit an integrity-violating transaction –or batch of grouped transactions– is not only useless –as the database will always raise the integrity exception– but also prevents the protocol from proceeding normally, stopping the processing of all transactions in the system. This cannot be avoided even though some other optimisations are used, such as the *concurrent writeset application* technique presented in [10]. With such optimisation, several non-conflicting transactions can be sent to the database, in such a way that the commit order can be altered from one node to the others if a transaction commits before a previously delivered one. This way, when indefinitely reattempting one integrity-violating transaction, subsequent non-conflicting transactions can be sent to the database, not stopping the processing of the node. However, this optimisation cannot be applied when the next transaction presents conflicts with the ones already sent to the database, so the protocol will stop all processing eventually.

5 Conclusions

None of the papers we have found deals with the problem of coordinating integrity checking with replication protocols. However, on the protocol level of replicated database architectures, many problems remain to be solved for implementing mechanisms that control transaction consistency, replication consistency and integrity, i.e., semantic consistency. One of them is addressed here.

We have presented an experimental study of the negative effects of not correctly managing integrity constraints. This has been accomplished by comparing the behaviour of two protocols. One of them reflected the traditional behaviour of protocols which do not care about integrity maintenance, based on the uncautious assumption that all transactions are programmed in such a way that they will preserve integrity. As opposed to that, the other protocol studied in our analysis properly handles semantic consistency as declared by integrity constraints. Only a simple modification of the traditional protocol is enough to reach the correctness of the second protocol. We have showed that an improper processing of integrity-violating transactions entails a history list that does not reflect the transactions actually applied in the database, which leads to errors when certifying subsequent transactions. This causes not only incorrect abortions but also incorrect commits. Moreover, resulting from the errors mentioned above, incorrect nodes present higher conflict-related abortion rates.

Finally, results also show that the proposed integrity-aware protocol introduces higher delays due to the larger extension of the mutually exclusive zone needed to safely access the history list. So, a correct integrity management introduces important delays that should start new research works looking for new performance optimisations compatible with such correct management.

References

1. Bernstein, P.A., Hadzilacos, V., Goodman, N.: *Concurrency Control and Recovery in Database Systems*. Addison Wesley, Reading (1987)
2. Gray, J., Helland, P., O'Neil, P.E., Shasha, D.: The dangers of replication and a solution. In: *SIGMOD*, pp. 173–182 (1996)
3. Wiesmann, M., Schiper, A.: Comparison of database replication techniques based on total order broadcast. *IEEE Trans. Knowl. Data Eng* 17(4), 551–566 (2005)
4. Gray, J.: Notes on database operating systems. In: In Bayer, R., Graham, R., Seegmuller, G. (eds.) *Operating Systems: An Advanced Course*, Springer, Heidelberg (1979)
5. Zuikeviciute, V., Pedone, F.: Revisiting the database state machine approach. In: *VLDB Workshop on Design, Implementation, and Deployment of Database Replication*, Trondheim, Norway, pp. 1–8 (August 2005)
6. Lafue, G.M.E.: Semantic integrity dependencies and delayed integrity checking. In: *VLDB*, Mexico City, Mexico, pp. 292–299 (September 1982)
7. Cammarata, S.J., Ramachandra, P., Shane, D.: Extending a relational database with deferred referential integrity checking and intelligent joins. In: *SIGMOD*, Portland, Oregon, pp. 88–97 (May 1989)
8. Liribat, F., Simon, E., Tombroff, D.: Using versions in update transactions: Application to integrity checking. In: *VLDB*, Athens, Greece, pp. 96–105 (August 1997)
9. Muñoz-Escóí, F.D., Ruiz-Fuertes, M.I., Decker, H., Armendáriz-Íñigo, J.E., González de Mendivil, J.R.: Extending Middleware Protocols for Database Replication with Integrity Support. In: *DOA*, Monterrey, Mexico (November 2008)
10. Lin, Y., Kemme, B., Patiño-Martínez, M., Jiménez-Peris, R.: Middleware based data replication providing snapshot isolation. In: *SIGMOD*, pp. 419–430 (2005)
11. Berenson, H., Bernstein, P., Gray, J., Melton, J., O'Neil, E., O'Neil, P.: A critique of ANSI SQL isolation levels. In: *SIGMOD*, San José, CA, USA, pp. 1–10 (May 1995)
12. Elnikety, S., Zwaenepoel, W., Pedone, F.: Database replication using generalized snapshot isolation. In: *SRDS*, Orlando, FL, USA, pp. 73–84 (October 2005)
13. Elnikety, S., Dropsho, S.G., Pedone, F.: Tashkent: uniting durability with transaction ordering for high-performance scalable database replication. In: *EuroSys*, Leuven, Belgium, pp. 117–130 (April 2006)
14. Irún-Briz, L., Decker, H., de Juan-Marín, R., Castro-Company, F., Armendáriz-Íñigo, J.E., Muñoz-Escóí, F.D.: MADIS: A slim middleware for database replication. In: Cunha, J.C., Medeiros, P.D. (eds.) *Euro-Par 2005*, vol. 3648, pp. 349–359. Springer, Heidelberg (2005)
15. Spread (2008), <http://www.spread.org>
16. PostgreSQL: (2008), <http://www.postgresql.org>

SEMELS 2008 PC Co-chairs' Message

We would like to welcome you – also in the name of the program committee members to whom we are very thankful for their work – to the First International Workshop on Semantic Extensions to Middleware: Enabling Large Scale Knowledge Applications. The workshop is to be held in conjunction with the On The Move Federated Conferences and Workshops 2008. With this workshop we will bring together researchers and practitioners from academia and industry working on innovative approaches to middleware systems that support semantically-enabled application scenarios or that exploit semantic technologies in order to improve their quality of service.

In large-scale systems, data and program heterogeneity have been a classical problem, to which semantic technologies are often presented as a solution. The classical solution to non-semantic integration has been middleware systems, which provide functionality for data and process mediation, coordination and composition. As semantics become more of a part of the underlying data and process layers of large-scale systems, extensions are also required in the middleware layer to not only support knowledge mediation and coordination but to provide new functionalities not previously possible due to the introduction of machine processability of the data and processes being integrated at the middleware.

As knowledge is becoming more and more ubiquitous in the Internet, we expect in the near future millions, even billions of knowledge and service providers and consumers to interact, integrate and coordinate on the emerging, open Semantic Web just as today millions of clients exchange data and perform work over the existing Web infrastructure. This produces the need for scalable and dynamic systems that support collaborative work with distributed and heterogeneous knowledge.

Such semantically extended middleware can be applied in any context which involves large-scale knowledge-based collaboration, such as in the biomedical and life sciences field or emergency planning, as well as facilitate new forms of intelligentWeb-scale coordination of processes (a step towards aWeb of services) which are currently not envisioned precisely because such a semantically-capable middleware layer is not existent.

At this first SEMELS workshop, we have the pleasure to present papers that augment very traditional infrastructures like P2P, Enterprise Service Buses (ESBs) or Tuplespaces with semantics, in order to enhance the aspects of life cycle management, configuration, scalability, or flexibility and adaptability.

The workshop moreover presents papers that exploit the application of semantic extensions of middleware to life science, healthcare or emergency management applications. In these application areas the use of semantic technologies shows some of the most advanced progress. Researchers in these domains deal with large-scale data integration and service cooperation on all levels and at all stages of their work. This more application-driven view on the requirements and solutions to semantic middleware approaches allows for advanced insights in the work of related fields. This is an

important issue on the way towards semantic extensions to middleware for large and complex networked knowledge systems.

November 2008

Reto Kruppenacher
Elena Simperl

Towards Semantically Enhanced Peer-to-Peer File-Sharing

Alan Davoust and Babak Esfandiari

Department of Systems and Computer Engineering
Carleton University
1125 Colonel By Drive
Ottawa, Ontario, Canada

Abstract. We characterize publication and retrieval of documents in peer-to-peer file-sharing systems and contrast them with query answering in peer-to-peer database systems. We show that the simplicity of file-sharing systems avoids many problems faced by P2P database systems. We propose a simple and open meta-model for documents and meta-data, for the purpose of expressing arbitrary relations between documents, peers, and file-sharing communities. Such relations in effect define a semantic enhancement to P2P file-sharing systems and enable the distributed emergence of knowledge. We illustrate our study with the description of our system, which distributes queries only to relevant peers, and can translate queries across different meta-data schemas.

Keywords: Peer-to-peer, file-sharing, Semantic Web.

1 Introduction

P2P file-sharing networks are groups of peers who make files available on their systems, and download those files from one another through peer-to-peer connections. Early systems include Napster [2], Gnutella [1], which allow simple keyword search, or basic metadata filtering.

P2P database systems (such as PIAZZA [3]), or “Peer Data Management Systems” (PDMS), are primarily database systems, distributed in a P2P manner, that collaborate to answer expressive queries. Such systems are meant to distribute queries to peers connected in arbitrary topologies, with heterogeneous schemas.

We will argue here these two types of systems, while having some conceptual similarities, are built on different assumptions and serve different purposes.

We are interested in discovering and accessing various *file-sharing* networks, with heterogeneous schemas and arbitrary topologies. Our hypothesis is that the dynamic and unstructured networks that support P2P file-sharing are ill-suited for processing highly expressive queries, which has been the central challenge in recent PDMS literature.

We believe that the simpler queries and constraints of the file-sharing setting (e.g. no issues with consistency or complete query answering) will make our

system more robust to a collective and truly decentralized contributions of data. Along with relations such as schema mappings, we propose to accommodate new types of relations between schemas (e.g. mappings with different semantics) or even between documents (e.g. successive versions of the same document).

In this paper we first analyze the defining features of file-sharing systems, modeling them in terms of database-type systems. In section 3 we introduce our meta-model of file-sharing communities. In section 4 we show how this model can manage queries across heterogeneous schemas, and finally we present a proposal to further enhance our prototype's support of data semantics, in section 5.

2 A Database Model for File-Sharing

Many file-sharing networks focus on specific data types (music, programs...), and searches support simple meta-data filtering. Such searches can be seen as relational *select* operations in a virtual database represented by the collection of shared files. Based on this analogy, we discuss here the main differences and limitations of file-sharing systems as compared with PDMS.

2.1 Database Schema

The conceptual database design of a file-sharing system only has one table, where the rows are documents distributed over the network. The columns are properties of each file, and include at least the two following fields:

- A unique identifier of the file, comprising its location on the network and its filename. This URL defines a primary key of the database table.
- The binary “payload” of the file, if there is any (e.g. the music in the mp3 file). We will call this the *data* field.

In addition, files may be described by additional meta-data fields, which define additional columns. For example, music files may be described by their artist name, song title, bitrate...

The database schema defines how these fields are encoded into the file itself. In traditional P2P file-sharing systems, it is hard-coded into the application.

It is important to note that the conceptual database schema cannot express any integrity constraints, apart from the implicit primary key constraint of the URL. As this constraint is de facto enforced by the network and filesystem properties, file-sharing systems are free of *data consistency* issues.

2.2 CRUD Functionality

Of the CRUD features of a database system (Create, Retrieve, Update, Delete), only Retrieve is really supported over remote data, since users don't have “write”-control over each others' disks. In other words, each source database offers an interface to a local user, who may access all four functions, and an interface for remote connections by other peers, which only offers the Retrieve function.

Furthermore, the Retrieve feature only accepts queries of limited expressiveness, as we discuss in the next section, and the Create and Update features are limited with respect to the URL field: only a certain range of values are accepted (i.e local URLs).

2.3 Expressiveness of Queries

Using the formalism of [9], a general conjunctive query of arity n over a relational alphabet A can be written in the form: $\{x \mid \exists y, body_{cq}(x, y)\}$ where $x = (x_1, \dots, x_n)$ are the free variables of the query, $y = (y_1, \dots, y_n)$ are existentially quantified variables, and $body_{cq}$ is a conjunction of atomic predicates of A , involving variables of x , y , and constants.

In traditional file-sharing systems, queries are limited as follows:

- Our relational alphabet A is limited to the atomic predicate T of arity p modeling the single database table of our schema.
- $body_{cq}$ only contains a single instance of T , and possibly conditions over the variables and constants (e.g. $x_1 \leq 10$).
- The predicate T appears in $body_{cq}$ as : $T(t_1, \dots, t_p)$ where t_i can be either x_i , y_i , or a constant. In other words, the query returns a contiguous subset of columns of the table T (intersected with an arbitrary subset of rows).
- In practice a P2P file-sharing system will accept two predefined types of queries:
 - *search* queries: the arity s of search queries is a constant $1 \leq s < p$: the query returns as a minimum the URL field, and excludes the data field.
 - *download* queries: the arity of the query is then $n = p$, and the tuples returned are entire files. The $body_{cq}$ is a single predicate of the form $T(a, x_2, \dots, x_p)$ where a is a constant URL.

In relational terms, these restrictions mean that *join* operations are not supported (a single instance of T is allowed in the query body) and that *project* operations are restricted to two predefined forms, corresponding to searches and downloads. This restricted granularity is due to the nature of the data: a file-sharing system is not a closed system, and manipulates primarily files or file URLs, which are pieces of data with a “well-known” semantics, and usable outside the system (as opposed to arbitrary tuples).

We note that leaving out *join* operations is important as it greatly simplifies query answering. Answering *join* queries in a fully distributed setting requires rewriting and executing queries according to query plans, which requires a certain amount of synchronization and collaboration at the protocol level, typically not provided by P2P file-sharing systems.

2.4 Expected Answers

The file-sharing purpose also implies that users may have different expectations with respect to the answers to their queries: complete and correct answers are

expected in the case of database systems, but more likely only a number of relevant answers are deemed sufficient in the case of file-sharing.

This implies that the *completeness* of query-answering algorithms in this setting is less important a requirement than fast answers, and justifies the fact that the propagation of searches in P2P networks can be limited to a certain depth, as in the Gnutella protocol.

Furthermore, as the data is replicated through the network via downloads, we expect popular data to spread and eventually become reachable to any peer, which can mitigate the effect of limiting the search depth.

2.5 Definitions

To summarize this analysis, we propose the following definitions for a P2P file-sharing system:

- A *P2P file-sharing community* is a set of peers using a common schema, defined by its name, and its interfaces for manipulating data (notably network protocol, specific format and syntax of built-in queries, rendering of the data to a human user...). Traditional file-sharing applications such as Napster define such communities.
- A *peer* is defined by a unique identifier in the network, and stores a local database, with a schema comprising a single table T , as defined in section 2.1.
- Each peer offers an interface to its local database with the following functions (in brackets, the input of the function):
 - *publish* (d : document): creates a new row in T containing the different fields of d . d must be stored locally (cf. section 2.2).
 - *search* (q : query): executes the query q over T and returns the matching tuples. The query q is of the form defined in section 2.3.
 - *download* [*server*] (q : query): executes q over T , and returns a document. As defined in section 2.3, q will match a single tuple, which is an entire document.
 - *remove* (d : document): removes a row from T .
- Peers can be accessed by a local (human) user or by a remote peer on behalf of its own local user. The functions *publish* and *remove* are accessible to the local user only.
- In addition to this database interface, the local user also has access to a high-level function *download* [*client*] (q : query) which encapsulates a request to a remote peer's *download* [*server*] function, and a *publish* of the returned document. In order to comply with the restrictions defined in 2.2, it is a local copy of the document that is published.

3 A Meta-model for P2P File-sharing Communities

In previous work [5,6], we proposed a meta-model for file-sharing communities consistent with this formalization, but making the community definitions ex-

plicit. In the following, we will base our discussion on our prototype system, called U-P2P, which implements the meta-model.

3.1 Meta-model

We consider a class of file-sharing communities, where the data being shared is stored in XML files, and the data field (as defined in section 2.1) is a separate unstructured file, linked by a URL. This format can accommodate any type of data and any format of meta-data.

In addition, resources are identified by a *resource-id*, which is implemented by a MD5-digest of the XML document. This identifier is meant to identify multiple copies of the same resource, which will have different URLs.

By describing the different parameters of a community in a structured document, we obtain a new resource that can be shared in a community of its own. For this purpose, we can introduce a special bootstrap community, the *community community*, or *root community*, to share documents in which community instances are defined.

An interesting aspect of the “shared in” relation is that the *root community* is related to itself : it can be defined in a structured document similar to that of other communities, and can be shared in the root community itself, thus bootstrapping the meta-model.

Figure 1 summarizes our meta-model and its difference with the model of a traditional file-sharing system.

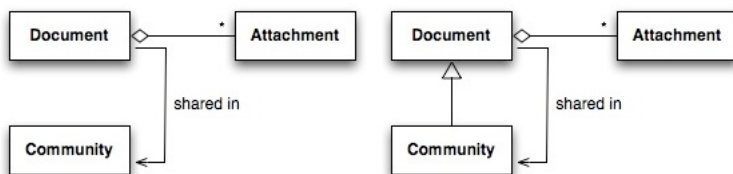


Fig. 1. Conceptual meta-model of a traditional file-sharing system (left) and U-P2P (right). In U-P2P, communities are represented as documents.

3.2 Active Documents to Handle Community Information

A benefit of this metamodel is that users can define a community simply by writing an XML document, or discover new ones by querying the *root community*. By downloading these *community* documents, their local system automatically acquires the capability of handling a new type of documents.

This last step is key to our approach. All the information defining the P2P community is shared at the data level, which is the only level at which peers communicate. To serve as a community, it must be interpreted by the implementing system. Attributes of the community, defined in the XML community documents, include attachments such as XSLT stylesheets defining how to render documents shared in this community.

Hence the community definition document, containing data but also encoding *behavior*, can be considered to be an *active XML document*, as defined by [8].

It is important to notice that from the user's high-level perspective, the system now supports more elaborate features, but these are built only with the basic file-sharing queries and protocols (such as Gnutella). We now have the building blocks of a complex P2P file-sharing client: documents, shared over simple file-sharing infrastructure, become active and enhance the system's high-level functionalities, e.g. by providing the capability of interpreting and viewing new types of documents.

In the next section, we show how more advanced semantic enhancements can be incorporated to the system, following the same approach.

4 Heterogeneous Data Representations, Data Integration

One of the central challenges studied in PDMS is to process queries across heterogeneous data schemata.

In the formalization commonly used in P2P data integration literature [9], peers advertise the *peer schema*, over which they accept queries. They then store *mappings* to the schemas of other peers, which are logical connections between two different data representations, describing their semantic relation.

In our setting, peers may simultaneously manage several schemas. It appears useful, as a preliminary analysis, to define the logical and physical architecture of a distributed system.

4.1 Physical vs. Logical Architecture

The term "Peer-to-peer" defines a decentralized interaction of peers over a network, and gives an indication of the network topology (i.e. it may be an arbitrary graph). This topology defines the *physical architecture* of the distributed system.

In the case of logical mappings between schemas, we can define in the same way a *logical architecture* for the distributed system, as the topology of the graph formed by the different schemata (nodes) and mappings (vertices). A system's logical and physical architectures may be different.

Recent work in P2P Data Integration has focused on systems where the logical architecture is that of an arbitrary graph, which adds many additional challenges to those of physical distribution.

Briefly, we note that systems such as PIAZZA [3] implement the most general architecture, a P2P logical architecture over a P2P physical network (the physical and logical graphs match), Edutella [4] implements a P2P (super-peer) physical architecture, but a centralized mediator-type logical architecture, where the network of super-peers share a common data model and distribute queries to "end-point" peers with different schemata.

Bibster [7] also has a centralized logical architecture, i.e. a core ontology to which all other data formats are mapped. Such systems are scalable to many physical peers, but potentially difficult to maintain: a small change in the central data-model can entail changes to all the other mapped models.

U-P2P allows each peer to interact with several different communities in a single physical peer, making the physical and logical architecture of the network very different. As we will discuss further, physical peers can then exploit this property to make logical connections in between two locally stored communities. Managing a logical connection within a single peer does not require very expressive communication at protocol level, i.e. between distinct physical peers.

4.2 Case Study

Following our previous approach, we have defined schema mappings in U-P2P, which we called “bridges”, in documents to be shared at the data level, and to be incorporated into the higher-level processing.

A first proof-of-concept case study was presented in [6]. In this work, two communities shared documents described in the meta-data formats of respectively the Fedora and DSpace digital libraries.

By downloading these two communities, a peer could completely acquire the capability of managing data in the two digital libraries and, in addition, acquire the capability of automatically translating searches from one community to the other, by downloading the relevant bridge. The peer would perform the following steps:

1. Download the Dspace and Fedora communities in the root community.
2. Search for bridges originating in the DSpace community.
3. Download the bridge linking these communities. By doing this, the client automatically acquires the functionality of searching across both communities.
4. Issue a search within the Dspace community. The user interface provides a link to extend the search to the Fedora community.
5. By clicking on a single link, the original DSpace query is automatically translated and sent in the Fedora community. The user thus obtains documents from both communities.

In this process, a peer issues a succession of simple file-sharing queries to its physical neighbors. By retrieving bridges, it can then extend queries across logical “hops”. The full process could be automated and made transparent to the user, which will become necessary with our proposed extension to arbitrary relations (cf. section 5).

4.3 Meta-model for U-P2P with Bridges

In the example above we introduced *bridges*. The schema for a basic bridge is: <source id> <bridge type> <destination id>. In this case-study, we only considered a single *bridge type*: this bridge represents a mapping from one community, identified by the *source id*, to the schema of the community identified by the *destination-id*. These identifiers are the ones introduced in section 3.1. Figure 2 shows the resulting meta-model.

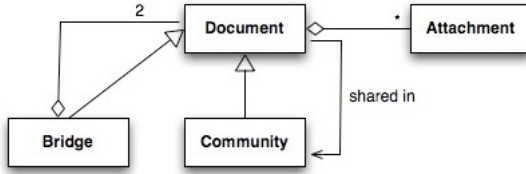


Fig. 2. Bridges in the U-P2P Meta-Model

5 Towards Bridges with Arbitrary Semantics

5.1 Bridges and RDF

The bridges defined in the previous section have a clear analogy with the schema mappings used in data integration, but also with RDF triples. Our P2P definitions described in section 3 can also be identified with RDF, as summarized in table 1.

Table 1. Correspondence between the U-P2P metamodel and RDF(S)

U-P2P Concept	RDF Concept
U-P2P Document	rdfs:Resource
U-P2P Community	rdfs:Class
U-P2P Bridge	rdf: Property
“shared in”	rdf:type

If a bridge can be identified to a relation in between two resources, the bridge *type* introduced in section 4.3 represents the *semantics* of the relation, which must be used by the system, for example to translate queries, as in the case study discussed above.

Just as community instances are interpreted by the system according to the semantics of the *root community* (cf. section 3.2), each instance of a given bridge *type* will be interpreted according to the semantics of its community.

Our approach, still in the prototyping phase, is to provide a framework for defining new bridge communities associated to new types of relations in between resources. We must describe the semantics of each abstract relation that we define, thus encoding new *behaviors* in the communities, to be interpreted using each bridge instance downloaded by the system.

Many relations would be useful, as the following example scenarios suggest:

- Scenario 1: Suppose a user wants to share descriptions of science papers, and finds an existing community of science papers, called “Publications”. The user also wants to include his own summary of each paper in the meta-data. The user could create another community called “SummarizedPublications”, which could *extend* the community “Publications”.

The “extend” relation could be defined as a bridge between two communities C_1 and C_2 , such that documents of C_2 contain an XML block complying

with the schema of C_1 , to which is appended another XML block complying with the schema associated with C_2 . The stylesheets associated with the community C_2 would be written to manage this format.

- Scenario 2: Suppose a user finds an error in a document downloaded from U-P2P and wishes to correct that error for the benefit of the other users. Making changes to that particular document and publishing it won't have much of an effect: one would need to make changes to all copies of the document. Our proposed alternative is to create a bridge, say `u-p2p:priorVersion`, between the erroneous copy and the new version.

5.2 Emergent Query Processing in a Tuple-Space Architecture

The solution we envision for implementing arbitrary bridges in our prototype system relies on a tuple-space architecture. Tuple-spaces are middlewares which allow agents to interact without explicit knowledge of one another. As the bridges should be freely added and removed from the system, a tuple-space is a good solution to decouple the execution of bridges from each other, to allow for runtime addition of new bridge types as well as dynamic coordination among them.

Agents will listen for tuples matching specific patterns and transform them according to their functionality, for example:

- A User-Interface agent will output “Query” tuples on behalf of the human user, and listen for “Query-Result” tuples.
- A Database agent will match any “Query” tuple and output “Query-result” tuples from the database.
- An agent representing the “priorVersion” bridge defined in scenario 2 above, will listen for “Query-result” tuples, and recursively output new queries for new versions of the document. A high-level behavior will emerge: the transitive closure of the “priorVersion” bridge is obtained.

We expect a collective behavior to emerge, and results to a given query will depend on which bridges are present on a user's system at the time of the query, defining a personalized and emergent semantics to queries.

In addition, we expect a second level of emergence to take place. As we assume bridges to be user-created, we must assume that some may be wrongly defined, or simply useless. Users may download bridges, delete them from their system, or on the contrary recommend them to others. Duplicates of bridges, as of any resource, may and will exist, only distinguishable by their location. The number of duplicates will be an indication of the “popularity” or usefulness of a given bridge. We conjecture that incorrect or needless bridges will disappear from the P2P network the way an unpopular song would disappear from Napster.

6 Conclusion

We have modeled simple file-sharing systems using a database model, and presented a line of research to increase the high-level capabilities of such systems while complying with their restricted expressiveness at protocol level.

We have shown how our prototype system U-P2P copes with these limitations, particularly the absence of explicit database schema support, by relying on active XML resources to carry relevant schema-level data, as well as attachments defining data-handling directives. Our underlying document meta-model nicely brings all data on the same level, documents, the communities that they are shared in, and relations in between all these documents.

Ongoing development aims to further enhance the support of user-defined relations in between documents, and exploit the “Darwinian” property of file-sharing systems to see complex high-level behaviors emerge from the distribution of active documents.

This analysis, as well as previously presented proof-of-concept examples, show the potential of this approach.

References

1. Gnutella protocol (retrieved on 07/07/2008), http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf
2. Napster file-sharing system (no longer exists), <http://www.napster.com>
3. Halevy, A., Ives, Z., Madhavan, J., Mork, P., Suci, D., Tatarinov, I.: The piazza peer-data management system. *Transactions on Knowledge and Data Engineering, Special Issue on Peer-data management* (2004)
4. Nejd, W., Wolf, B., Changtao, Q., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmer, M., Risch, T.: Edutella: A P2P networking infrastructure based on RDF. In: *Proceedings of WWW 2002*, ACM Press, New York (2002)
5. Arthorne, N., Esfandiari, B., Mukherjee, A.: U-P2P: A Peer-to-Peer Framework for Universal Resource Sharing and Discovery. In: *USENIX 2003 Annual Technical Conference, FREENIX Track*, pp. 29–38 (2003)
6. Arthorne, N., Esfandiari, B.: Peer-to-peer Data Integration with Distributed Bridges. In: *Proceedings of the 2006 conference of the Centre for Advanced Studies on Collaborative Research*, pp. 174–188 (2006)
7. Broekstra, J., Ehrig, M., Haase, P., Van Harmelen, F., Mika, P., Schnizler, B., Siebes, R.: Bibster - a semantics-based bibliographic peer-to-peer system. In: *Proceedings of the Third International Semantic Web Conference*, pp. 122–136 (2004)
8. Ciancarini, P., Tolksdorf, R., Zambonelli, F.: Coordination Middleware for XML-Centric Applications. In: *Proceedings of the 16th ACM Symposium on Applied Computing, Madrid (E)* (March 2002)
9. Calvanese, D., Giacomo, G., Lenzerini, M., Rosati, R.: Logical foundations of peer-to-peer data integration. In: *Proceedings of PODS 2004*, pp. 241–251 (2004)

Efficient Content Location in Massively Distributed Triplespaces

Kia Teymourian and Lyndon Nixon

Free University of Berlin, Institut for Computer Science, AG Networked Information Systems
Königin-Luise-Str. 24/26, 14195 Berlin, Germany
{kia,nixon}@inf.fu-berlin.de
<http://nbi.inf.fu-berlin.de>

Abstract. Triple Space Computing is a new middleware paradigm [1][3] based on semantics and tuplespaces which can be used for the coordination of Semantic Web clients and services. To achieve scalability of Triple Space infrastructure distribution of triplespaces is necessary. A major problem within massively distributed triplespaces is to find the best suited triplespaces to answer a certain query. In this paper we introduce a novel approach for efficient content location of triplespaces given a certain query. We use a Peer-to-Peer overlay based on Distributed Hash Tables and three semantic overlay layers which are used to score the known triplespaces according to their probability to answer queries. This combination approach is introduced to solve the triplespace selection problem more efficiently and support high performance operation handling in triplespaces [4].

1 Motivation

Tuplespace-based computing [6] is a powerful concept for the coordination of autonomous processes. Instead of explicitly exchanging messages or performing remote procedure calls, inter-process communication is performed by reading and writing distributed data structures in a common virtual space connecting distributed entities. Coordination is carried out on a data-driven basis, in that operations attempting to access data in the space are blocked until the denoted data becomes available. Spaces have application to Web-based communication in that they realize open places where information can be published and persistently stored. They have advantages over the standard client-server model in cases of concurrent processing of published information by heterogeneous sources. This has been successfully demonstrated in space-based systems applied to solve communication and coordination issues in areas such as open distributed systems, workflow execution and XML middleware [2][3][4]. Here, applications were able to collaborate on tasks independent of their individual implementations, or whether a communication partner was online. We have proposed a conceptual model for such a communication middleware previously [12]. This middleware (“Triple Space”) manages information formalized using Semantic Web representation languages and coordinates the exchange among distributed entities (such as Semantic Web services) that process this information. In this paper we return to Triple Space with a focus on distribution

¹ This work is funded by the European Commission under the project TripCom (IST-4-027324-STP).

and efficient content location in triplespaces. Firstly, we place this within the context of the relevant state of the art. Then we present our chosen architecture and provide some additional detail on our API operations. Finally, we describe in detail our approach for distribution in triplespaces. To conclude, we focus on planned future work.

2 Related Works

Some work has already been achieved in the area of semantic tuplespaces. This prior work feeds directly into our work on Triple Space. Triple Space Computing was first described in [5]. The work was elaborated in the scope of the Triple Space Computing (TSC) project.² Conceptual Spaces/CSpaces was born as an independent initiative to extend Triple Space Computing [5] with more sophisticated features and to study their applicability in different scenarios apart from Web services. This work was not fully completed and its purpose too heavy-weight for our intended application as a "lighter" middleware layer for semantic clients. Semantic Web Spaces [17] has been envisaged as a middleware for the Semantic Web, enabling clients using Semantic Web data to access and process knowledge to co-ordinate their interdependent activities. Semantic Web Spaces takes a lightweight approach to implementation to allow for more flexibility in the selection of solutions for different aspects of the implementation. Some of the experience from Semantic Web Spaces has been brought to the modelling of Triple Space. An important factor here is that the aforementioned semantic tuplespaces are based on centralized implementations, which cannot scale and do not fit well to the intention of providing a middleware for the Semantic Web and Semantic Web Services.

Related work to our distribution approach can be split into distributed tuplespace implementations and distributed semantic repositories. Relevant tuplespace implementations which are distributed, but do not support semantic data are Gigaspaces, PeerSpace, Blossom, DTuples, SwarmLinda, Grinda, Lime, Comet and Tota. Recent work has focused on either hash-based or self-organizing strategies. In the former, the performance/scalability to be expected can be inferred from P2P-based hash table distribution results. Semantic information has been distributed in this fashion in distributed RDF repositories like Edutella, YARS, RDFPeers, PAGE and GridVine. Systems already demonstrate good performance results. At least for simple queries, scalability is also promising. They do not however support the coordination functionality of tuplespaces. In the latter, there are promising early results for small amounts of different tuple types [7]. Since the distribution of semantic tuples (triples) may be less simple to divide into a small number of clearly delineated types, it is not yet proven if self-organization with semantic tuples can be as effective. Additionally, security constraints in Triple Space prevent the use of swarm-like clustering of tuples in the entire network.

3 Architecture of Triple Space

The Triple Space (TS) infrastructure is realized by a multitude of Triple Space kernels which clients can connect to as shown in figure 1. Each kernel can be identified with a

² <http://tsc.deri.at>, funded by the Austrian Federal Ministry of Transport, Innovation and Technology under the FIT-IT Semantic Systems action line.

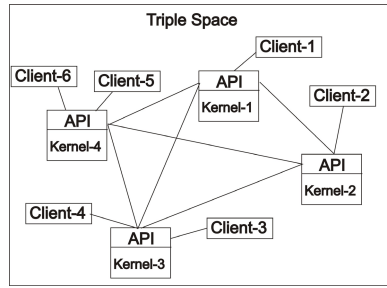


Fig. 1. Triple Space Infrastructure

URL. (For example kernel `tsc://k1.inf.fu-berlin.de:2588`) The Triple Space kernel consists of a number of self contained components that deliver jointly the Triple Space functionality [16]. A triplespace is identified by a unique identifier and managed by a kernel. It is accessed through Triple Space API operations as described in section 4. Reasoning occurs locally on kernels, i.e. inferences are calculated in the storage layer of each kernel based on space content - both instance data and the associated ontologies. More about the architecture and implementation can be found at [16].

Each kernel includes a component which is called *Distribution Manager* which has the task to look up kernels and route operations to other kernels and collect the responses for clients.

4 Triple Space API

Users of Triple Space use Triple Space clients to access kernels for sending and receiving data throughout the entire Triple Space using the provided Triple Space API operations. We defined 18 API operations and categorized them into 3 groups: Core, Extended and Further Extended. In this paper, we focus on the Core API operations which are defined as follows:

- *void Out(Triple t, URL space)*; writes a single triple into the space. The client is immediately free to perform further activities.
- *Set<Triple> rd(SingleTemplate t, URL space, Time timeout)*; Returns one match of the given template which is a single triple pattern. The match may be a set of triples, e.g. Concise Bounded Description. A timeout is provided to give a temporal bound for returning a match.
- *Set<Triple> rd(SingleTemplate t, Time timeout)*; As the *rd* operation above but no space URL is given. The system is free to select a match from anywhere in Triple Space where the client has read permissions. It is clear that the result of a read operation without space URL must not be the same as the read with the target space URL, because the system is here free to read the triples from any triplespace which can answer the query with the minimum latency.

The Extended and Further Extended APIs add further access primitives such as *rdmultiple* (retrieve all found matches for the query), *subscribe* (notify on a new match) and

in (deleting triples which match the query). These place additional requirements on the distribution implementation which will be the subject of future work.

Based on the fact that the users of Triple Space store triples into triplespaces using the API *out* operation which names a target triplespace URL, the provided triple must be stored to that specific kernel. In other words, the distribution of the emission of triples into Triple Space is controlled by the user; a triple is emitted to and remains in the given triplespace. Triple relocation between spaces is not possible due to security aspects, the workaround being of course a *in* which targets that triple and then *out*'s it into the new space.

Consequently, the main task of the distribution manager component of the kernels is not to distribute triples among spaces but to lookup triples which have been distributed in spaces, in particular to locate the responsible kernel and issue the queries and return the results to the clients.

5 Distribution Strategy

Distribution of the Triple Space infrastructure is based on strategies which are designed to handle the TS API operations. The first strategy is that each triplespace is identified by an URL and TS clients or kernels can use the Domain Name Services (DNS) of the World Wide Web to resolve the space URL to the IP address of the resident kernel. By each *out* operation, clients provide a space URL and specify the data (triples) that should be stored in that specific triplespace. Handling the read operations is more complex because they might provide no target space URL, then the kernel must look up potential spaces which can answer such a query. The second strategy is to create indices of the provided triples and their triplespaces. The Triple Space infrastructure includes a distributed index storage system which can store the indices of triples and triplespace URLs. This distributed index storage system is accessible for all of the Triple Space kernels. The final strategy is based on the local semantic knowledge of each kernel about other kernels in the network. We use semantic overlay layers for efficient kernel selection and ranking of kernels, and combined them with distributed hash tables to optimize the routing system to support high-performance operation handling.

5.1 Distributed Index Storage

Considering a triple (s,p,o) in which s, p and o indicate the subject, predicate and object of the RDF triple, we introduce the following index model: $\langle s, p, o, SpaceURL \rangle$. This index model is used as a data model for storage in the distributed index storage which is a distributed database built on top of a P2P overlay layer. Kernels are in a P2P relationship with each other and build up the peers in a P2P system. Each Distribution Manager includes a database instance of the distributed index storage system (Figure 2).

The P2P physical layer uses typically two parameters, the key and the data value. The key of a data item is generated from the data value using an order-preserving hash function. Each distribution manager in the system is responsible for storing the data values with keys which fall under its current key space. The physical storage layer supports two main operations, *Retrieve(key)* which can be used to search data items in the whole system and *Insert(key, value)* which is used for data storage, update and erasing. The fundamental idea of this approach for distribution in Triple Space is to store

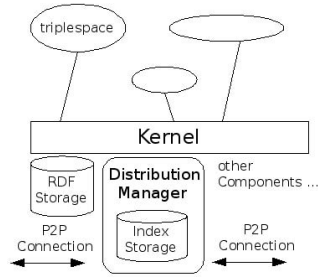


Fig. 2. Index Storage Component

the space URLs together with the triples in a distributed index storage system to be able to keep tracks of triples which are available in the Triple Space System. To achieve this issue, we suppose the following (Key, Value) pairs to be stored in the storage system.

- Key1 = Hash(s) Value = (s, p, o, SpaceURL)
- Key2 = Hash(p) Value = (s, p, o, SpaceURL)
- Key3 = Hash(o) Value = (s, p, o, SpaceURL)
- Key4 = Hash(SpaceURL) Value = (s, p, o, SpaceURL)

We investigated different P2P overlay implementations and came to the conclusion to use the P-Grid [11] overlay network, because it provides many facilities for self-organization and replication. We store our indexes in UniStore [8] which is a large-scale but still lightweight distributed data system on top of P-Grid. For efficiency reasons by query processing, we store triples with the space URL and don't divide the triple into individual indexes like $\langle s, \text{SpaceURL} \rangle$, $\langle p, \text{SpaceURL} \rangle$ and $\langle o, \text{SpaceURL} \rangle$. The retrieve operation is shown to locate a single match of the given triple template very efficiently, as there is no need for join operations on the indexes (only in the case of a binding of a variable to a blank node is it necessary to subsequently query the hosting kernel in order to acquire the triple pattern's *concise bounded description* [5]).

The distribution manager follows the following steps by each search on index storage:

1. Determine the keys from *bounded values* of the SPARQL query (basic triple patterns)
2. Send the retrieve request to the distributed index storage to get the index values. The P2P system routes this retrieve request to the corresponding kernel and its distribution manager, the results will be returned to the requested kernel.
3. Get a set of the space URLs
4. Forward the query to one of the spaces and its kernel.

The extended configuration of the Triple Space API aims to support *rd* operation with the full support of SPARQL language as template pattern. Distribution manager maps the provided SPARQL query to a SQL query upon the distributed index storage in order to lookup candidate triplespaces. It uses the OpenRDF [4] querying facility to parse

³ Defined in <http://www.w3.org/Submission/CBD/>

⁴ <http://www.openrdf.org>

Table 1. Shortcut Table for Triple Providers and Recommender Kernels

Set(TriplePatterns)	KernelURL	Hits	Type	Timestamp
((?s, rdf:type, ?o))	tsc://fu-berlin.de	43	T	7654478120
((?s, name, ?o),(?s, type, auc))	tsc://fu-berlin.de	23	T	2342478120
((?s, rdf:type, dam:cc))	tsc://inf.fu-berlin.de	12	R	1212478120

the SPARQL query and to create RDF basic triple patterns (set of triple patterns). The set of triple patterns is then used to create a SQL query for usage with the distributed index storage system. Joins need to be performed locally on the bindings acquired for the individual triple patterns from directly querying the hosting kernels (as inferrable bindings may also be of relevance).

5.2 Semantic Overlay Layers

It shall be noted in a scaled up Triple Space the set of candidate kernels for a triple pattern may grow to a size where considering all candidates is infeasible from the perspective of generated overhead. Also, triple patterns may not be matched in the distributed index storage because matches exist only in Triple Space as triples inferrable locally on kernels, hence a rule of thumb must be established for such triple patterns to find potential kernels which can have a match through inference.

Three semantic overlay layers are used to rank kernels according to their capability to handle query operations. We used the ideas of semantic overlay networks of existing work on semantic methods for P2P query routing [9], interest-based routing in [14] and semantic routing in [15]. We extend existing work by combining these approaches with our distributed index storage system and use them for supporting high-performance operation handling in distributed triplespaces. We use the following three shortcut creation strategies and create ranking metrics to select a kernel which has the best potential to answer queries.

Triple Provider and Recommender Layer. Triple provider kernels are kernels which could successfully answer queries in the past. Such kernels build the *Triple Provider Layer* for the local kernel. The local kernel creates shortcuts for such kernels and stores them in a local shortcut table. Shortcuts are simple logical links from the local kernel to the remote kernels. Each time the querying kernel receives an answer from a remote kernel, the following shortcuts Φ to remote kernels are added to the shortcut table of the local kernel. In the shortcuts, *Triple Pattern* refers to the RDF triple patterns which are given in the issued read operation, *Query Hits* are the number of successfully answered operations with the same query, *Type* is the type of shortcut. We store the triple provider shortcuts with type *T*. *Timestamp* is the timestamp of the latest successful operation with this query.

$$\Phi(\text{Set}(\text{Triple Pattern}), \text{Kernel URL}, \text{Query Hits}, \text{Type}, \text{Timestamp})$$

If no kernel is known to have answered the specific query we search for a kernel that issued the query in the past. We assume that this kernel has been successful in getting results and can forward the query to the target kernel, because it knows the triple

Table 2. Shortcut Table for Favorite Kernels

KernelURL	Shortcuts	Kernels	F. Factor
tsc://fu-berlin.de	20	30	600
tsc://inf.fu-berlin.de	32	10	320
tsc://mi.fu-berlin.de	15	20	300

provider kernel. Such kernels are recommender kernels and make up the *Recommender Layer*. Shortcuts for the Recommender Kernels are stored in the same table as the triple providers. The type of recommender kernels are “R”. Each kernel includes a shortcut table like the following table and stores shortcuts to remote kernels. With each successful routing, the local kernel can learn more about other kernels in the Triple Space network. Table 1 shows a sample table for shortcuts for triple providers and recommender kernels.

Only a limited number of the shortcuts will be kept in shortcut tables and based on the timestamp and the number of hits, shortcuts will be removed from the table over time. **Favorites Layer:** Active kernels issue many operations and produce many shortcuts. Kernels are especially interested to know which kernels in the network are most active and can answer queries more quickly. Each kernel creates shortcuts of type Ψ to its favorite kernels which are ranked by query activity.

$$\Psi(\text{Kernel URL}, \text{No. of Shortcuts}, \text{No. of Known Kernels}, \text{Favorite Factor})$$

Each kernel includes a table like table 2 for its favorite kernels. The favorite factor is computed from the number of shortcuts the kernel has created multiplied by the number of remote kernels it knows. With each change in the shortcut table, favorite shortcuts will be updated. The local kernel asks remote kernels for their number of known kernels at intervals to update the shortcut table for favorite kernels. Each kernel will only keep a limited number of top favorite kernels in its table, because only kernels with the best favorite factors will be used for operations. This will reduce the number of messages which are needed to keep the table up to date.

6 Storing and Indexing Triples

Triples can be stored in triplespaces using the *out* operation which includes the target triplespace. The physical network address of the kernel on which the space is hosted is resolvable from the space URL. In this way, the user’s client could build a connection directly to the Triple Space API of a specific kernel which is identifiable from the space URL and perform the *out* operation. If the client is already connected to a Triple Space kernel and wants to store triples in a non-local triplespace, the Distribution Manager of the local kernel can forward the *out* operation to the target kernel. The Distribution Manager performs the following steps by each *out* operation:

1. Check if the local kernel is responsible for this operation or if it should be forwarded to a remote kernel. In the case of forwarding, build a connection to the API of the remote kernel and forward the operation.

2. If the local kernel is responsible, check if the indexing for this operation is permitted. If yes, send the storage request for the index value (s, p, o, SpaceURL) to the distributed index storage system.
3. Store the triple in the target triplespace (for persistence, the triple is added to a RDF storage layer).

This strategy for storage of triples make it possible that the system is able to retrieve again the triple from anywhere using the keys generated from s, p and o. The indexing is only used for the lookup by retrieval operations and the RDF storage layer provides the reasoning facilities. We also introduce public and private triplespaces. In the case of a public triplespace, anyone can read its triples, and by private triplespaces the read may be restricted by some access policies. The distribution manager is only permitted to index triples in public triplespaces, making them available over the distributed index storage system. It is clear that the number of triples which are stored in the distributed index storage will be less than the total number of triples which are stored in the RDF storage of kernels.

7 Querying Triple Space

For querying triplespaces, the client can choose from two kinds of read operation. Either a target triplespace URL is given (and queried) or no URL is specified, in which the system decides where to search for a match to the query (anywhere in Triple Space). In this section we introduce our combinatorial approach for handling read operations.

Read Operations *with* Space URL: The distribution manager checks by each read operation, if the target space is on the local kernel or on a remote kernel. To forward a read operation to a remote kernel, distribution manager takes the triplespace URL and resolves it to the IP address of the kernel using DNS. The communication channel between the kernels is realized using the SOAP protocol and Web Services. Distribution manager uses a Web Service client to connect to the API implementation of a remote kernel. By each successful read operation a shortcut for the *Triple Provider* kernel will be created. The important side effect of this is that by each successful read with space URL which is forwarded to a remote kernel, the local kernel can create shortcuts and get gradually more knowledge about its network.

Read Operations *without* Space URL: In this case, any triplespace may be queried. The Distribution Manager performs the following steps:

1. Check the local triplespaces, if the query can be answered locally, return the results to the client. If not, go to the next step.
2. Check the shortcut table for the exact match of this query, if there is an exact match, forward the query to that remote kernel. If not, go to the next step.
3. If no shortcut exists for the this query, send the retrieve request to the distributed index storage to get the index values. The P2P system routes this retrieve request to the corresponding kernel and its distribution manager, the results will be returned to the local kernel. Result is a set of triplespace URLs and distribution manager forwards the query to these kernels, and with the first match, it sends the kill message (signal) to any of the started processes for this thread on local machine. If the distributed index storage system has no results for this query, go to the next step.

4. Forward the query to one of the top favorite kernels.

For all of these operations, the local read operation process will terminate after the timeout period is over. If the distribution manager can not find any shortcuts, it will use the default layer which is based on distributed hash tables. Forwarding of operations will happen through the API, this means that the distribution manager starts a process to connect to the API of the remote kernel and forward the operation to that kernel. It waits until it gets a result or the timeout is reached. To avoid cycles between the kernels by forwarding of operations, a list of already visited kernels is also attached and forwarded with the operations.

The above mentioned steps are used to look up kernels which can answer the SPARQL query of the read operation. After finding some potential kernels, we simply forward the query to one of the selected kernels, and the target kernel can process the SPARQL query on its local RDF storage. By *read with space*, we use the DNS, and forward the query to the target kernel. Hence, the query will be processed on one single kernel and not distributed on several kernels, while there are some ongoing research works on distributed query processing in Triple Space [10].

A final arising issue is about the virtual or inferred triples that do not exist explicitly and are results of some inference or reasoning operation on the RDF graph, can also be addressed with read operations on one single kernel using the reasoning facility of the local kernel. The read operation *with space URL* defines that the user starts a read operation and wants that this operation be processed on the named triplespace and its subspaces. The named kernel which host these triplespaces can include inferred triples in the result of query processing using the reasoning facility of the local kernel. By a read operation *without space*, the system is free to lookup target triplespaces, map the read operation to a *read with space* operation and start the operation on the found kernel. A distributed reasoning on the global and complete knowledge of the Triple Space infrastructure is not the subject of this project.

8 Future Works and Conclusion

The Triple Space middleware has been successfully used to implement two scenarios: an eHealth scenario, in which European Patient Summaries in RDF are shared across an European network for coordinated access by health professionals; and a Digital Assets Management scenario, in which digital asset metadata in RDF is negotiated by content providers and content users in an auction pattern.

For the distribution, we have integrated P-Grid and UniStore [18] with other approaches for semantic P2P query routing [9][14][15]. We replaced the default flooding layer of existing works with our indexing system which brings a significant improvement in reducing routing costs and latency.

Our next step is to benchmark the Triple Space implementation to formally prove its validity as a middleware for the Semantic Web and Semantic Web Services. The scalability evaluation will be performed on the Amazon Elastic Compute Cloud (EC2)⁵, which will allow for running up to 100 kernels. Test clients will connect to these kernels from outside the EC2. We also plan to further optimize the implementation by:

⁵ <http://aws.amazon.com/ec2>

distributing the persistent storage back-end, providing for self-organization principles in triplespace data by allowing spaces to distribute themselves over multiple kernels and by implementing concurrent distributed query processing.

References

1. Aberer, K., Cudré-Mauroux, P., Datta, A., Despotovic, Z., Hauswirth, M., Puceva, M., Schmidt, R.: P-grid: A self-organizing structured p2p system. *ACM SIGMOD Record* 32(2) (September 2003)
2. Cabri, G., Leonardi, L., Zambonelli, F.: MARS: a programmable coordination architecture for mobile agents. *IEEE Internet Computing* 4(4), 26–35 (2000)
3. Ciancarini, P., Knoche, A., Tolksdorf, R., Vitali, F.: PageSpace: An Architecture to Coordinate Distributed Applications on the Web. *Computer Networks and ISDN Systems* 28(7–11), 941–952 (1996)
4. Ciancarini, P., Tolksdorf, R., Zambonelli, F.: Coordination Middleware for XML-centric Applications. *Knowledge Engineering Review* 17(4), 389–405 (2003)
5. Fensel, D.: Triple-Space Computing: Semantic Web Services Based on Persistent Publication of Information. In: Aagesen, F.A., Anutariya, C., Wuwongse, V. (eds.) *INTELLCOMM 2004*, vol. 3283, pp. 43–53. Springer, Heidelberg (2004)
6. Gelernter, D.: Generative communication in linda. *ACM Trans. Program. Lang. Syst.* 7(1), 80–112 (1985)
7. Graff, D.: Implementation and Evaluation of a SWARMLINDA System. Technical Report TR-B-08-06, Free University of Berlin (June 2009)
8. Karnstedt, M., Sattler, K.-U., Richtarsky, M., Müller, J., Hauswirth, M., Schmidt, R., John, R.: UniStore: Querying a DHT-based Universal Storage (2006)
9. Löser, A., Staab, S., Tempich, C.: *Semantic Methods for P2P Query Routing*. Springer, Heidelberg (2005)
10. Obermeier, P., Nixon, L.: A Cost Model for Querying Distributed RDF-Repositories with SPARQL. In: *Proceedings of the Workshop on Advancing Reasoning on the Web: Scalability and Commonsense*. *CEUR Workshop Proceedings*, vol. 350
11. Shafiq, O., Krummenacher, R., Martin-Recuerda, F., Ding, Y., Fensel, D.: Triple space computing middleware for semantic web services. In: *EDOCW 2006: Proceedings of the 10th IEEE on International Enterprise Distributed Object Computing Conference Workshops*, Washington, DC, USA, p. 15. IEEE Computer Society, Los Alamitos (2006)
12. Simperl, E., Krummenacher, R., Nixon, L.: A Coordination Model for Triplespace Computing. In: *9th Int'l Conference on Coordination Models and Languages* (2007)
13. Simperl, E.P.B., Krummenacher, R., Nixon, L.J.B.: A coordination model for triplespace computing. In: *Murphy, A.L., Vitek, J. (eds.) COORDINATION 2007*. LNCS, vol. 4467, pp. 1–18. Springer, Heidelberg (2007)
14. Sripanidkulchai, K., Maggs, B., Zhang, H.: Efficient content location using interest-based locality in peer-to-peer systems. In: *Infocom*. IEEE (April 2003)
15. Tempich, C., Staab, S., Wranik, A.: Remindin': semantic query routing in peer-to-peer networks based on social metaphors. In: *WWW 2004: Proceedings of the 13th international conference on World Wide Web*. ACM, New York (2004)
16. Teymourian, K., Nixon, L., Wutke, D., Moritsch, H., Krummenacher, R., Kühn, E., Schreiber, C.: Implementation of a novel Semantic Web middleware approach based on triplespaces. In: *Workshop on Middleware for the Semantic Web, ICSC 2008* (August 2008)
17. Tolksdorf, R., Paslaru Bontas, E., Nixon, L.: Towards a tuplespace-based middleware for the Semantic Web. In: *Proc. IEEE/WIC/ACM Int'l Conf. on Web Intelligence WI 2005*, pp. 338–344. IEEE Computer Society, Los Alamitos (2005)

Extending ESB for Semantic Web Services Understanding

Antonio J. Roa-Valverde and José F. Aldana-Montes

Universidad de Málaga, Departamento de Lenguajes y Ciencias de la Computación
Boulevard Louis Pasteur s/n 29071 Málaga Spain

{roa, jfam}@lcc.uma.es

<http://www.lcc.uma.es>

Abstract. The potential growth of applications distributed over a network and the large number of users has created the need for an infrastructure which can support increasing interaction not only among such users and applications but also an application to application interaction. This problem known as application integration has been addressed throughout time using different approaches such as Service Oriented Architecture (SOA), Enterprise Application Integration (EAI) and Web Services. The Enterprise Service Bus (ESB) draws the best traits from these and other technology trends. In this work, we propose the use of an ESB combined with Semantic Web technology with the aim of building an “intelligent” middleware which facilitates the Semantic Web Services life cycle and the deployment of new computation resources over the architecture.

1 Introduction

Nowadays, ESB has become the most promising solution to build a SOA infrastructure from several heterogeneous sources. Moreover, there are a lot of projects and work being carried out by researchers in the area of Semantic Web Services. However, many of those works have focused on the search for a suitable technology to model traditional Web Services applying the acquired knowledge in the Semantic Web. Proof of this fact can be seen in the number of proposals submitted to the W3C¹. Although there is not an official standard for modelling Semantic Web Services we believe that developed approaches must be put in practice. In this way it is possible to use Semantic Web Services in conjunction with an ESB to overcome the problem of enterprise integration.

The use of the Semantic Web Services technology in enterprises would not be possible without the existence of an infrastructure that allows covering the life cycle of Web Services using semantic annotation techniques. This necessary infrastructure could be an ESB, which would facilitate the integration of various heterogeneous systems. An ESB allows the cooperation and the exchange of data between services. It is a logical architecture based on the principles of

¹ <http://www.w3.org/2002/ws/swsig/>

SOA, which aims to define services explicitly and independently of the implementation details. It also pays close attention to securing a transparent location and excellent interoperability.

An ESB makes Web Services, XML, and other integration technologies immediately usable with the mature technology that exists today. The core tenets of SOA are vital to the success of a pervasive integration project, and are already implemented quite thoroughly in the ESB. The Web Service standards are heading in the right direction, but remain incomplete with respect to the enterprise-grade capabilities such as security, reliability, transaction management, and business process orchestration. The ESB is based on today's established standards in these areas, and has real implementations that are already being deployed across a number of industries. The ESB is quite capable of keeping in step with the ongoing evolution of the Web Services equivalents of these capabilities as they mature [1]. It would be interesting to maintain these capabilities over the use of Semantic Web Services.

In this paper we propose the implementation of an infrastructure composed of different layers where the ESB is the foundation on which the others are based. The objective is to define a Semantic Enterprise Service Bus (SESB), providing mechanisms to collect all these technologies together and acting as a layer to access services through the invocation paradigm based on goals. This idea relies on a combination of several standards from the field of Web Services and Semantic Web technologies within the ESB: SAWSDL [7], for creating semantically annotated descriptions of service interfaces, and OWL [8], for modelling relationships among deployable artifacts across the ESB and performing the integration process via DL reasoning.

The remaining of this paper is structured as follows. In Section 2, we introduce the motivation that guides this work. In Section 3, we describe the two possible approaches to build a Semantic Enterprise Service Bus. We also show a schematic vision of both architectures. In Section 4 we summarize current trends that aim to extend the use of Semantic Web Services, and Section 5 discusses future work and concludes this paper.

2 Motivation

For several years many approaches to overcome the application integration problem have been proposed, i.e. CORBA², EAI, ESB, etc. Despite, these approaches relying on different technologies and mechanisms they share a common point of view: software engineers are responsible for understanding the different application specifications and coordinating them to build a more complex system. Figure 1 depicts the necessary process to deploy a solution using an ESB. This process consists of two phases. Firstly, the software engineer must create the configuration file used for the ESB to initialize listeners in the startup phase. In this way, the software engineer must know with a high level of detail the different applications that he/she wants to integrate, i.e. accepted inputs and

² <http://www.corba.org/>

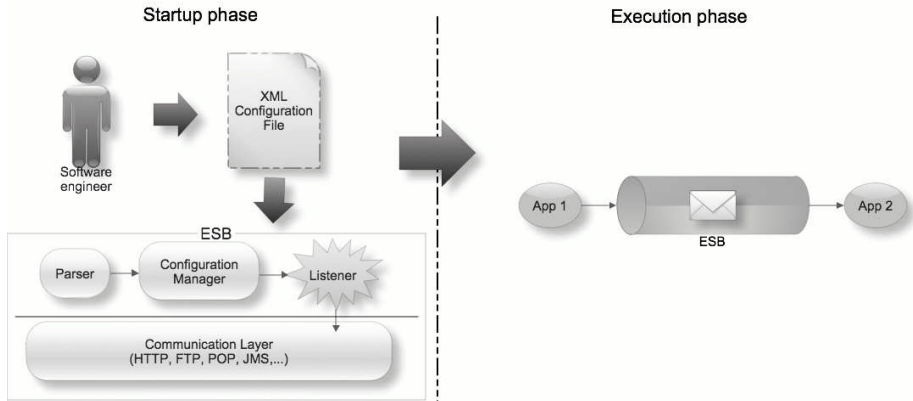


Fig. 1. Typical ESB usage

outputs, listener ports, protocols, etc. In the execution phase the ESB is ready to accept messages and transport them among applications using the information stored in the configuration file. As we can see, the entire process relies on the configuration file coded manually by the software engineer.

Until today, proposals have been focused on providing a middleware to solve heterogeneity and communication problems among applications without taking into account information relative to the meaning of the data that these applications can process. So, a tool capable of processing this kind of information would be very helpful for software engineers. Our aim relies on applying this idea to Semantic Web Services. In this way, a tool like this could facilitate frequent tasks in this field such as service composition and discovery. The desired idea tries to avoid writing the configuration file manually. We can imagine a software engineer trying to integrate several Semantic Web Services with the aim of building a more complex service in a composition process. Ideally, the software engineer could introduce the required goal and the ESB would be able to create the configuration file in an automatic or semi-automatic way using the available semantic annotations (see Figure 2).

3 SESB Architecture

SESB aims at providing developers with a middleware that facilitates application integration tasks through Semantic Web Service technology. There are two different ways to build such infrastructure using an ESB. The first one uses the ESB as the basis layer for building the topology on which different components are deployed. The second one tries to extend the ESB adding a new module responsible for understanding the semantic annotations over the artifacts deployed on the ESB.

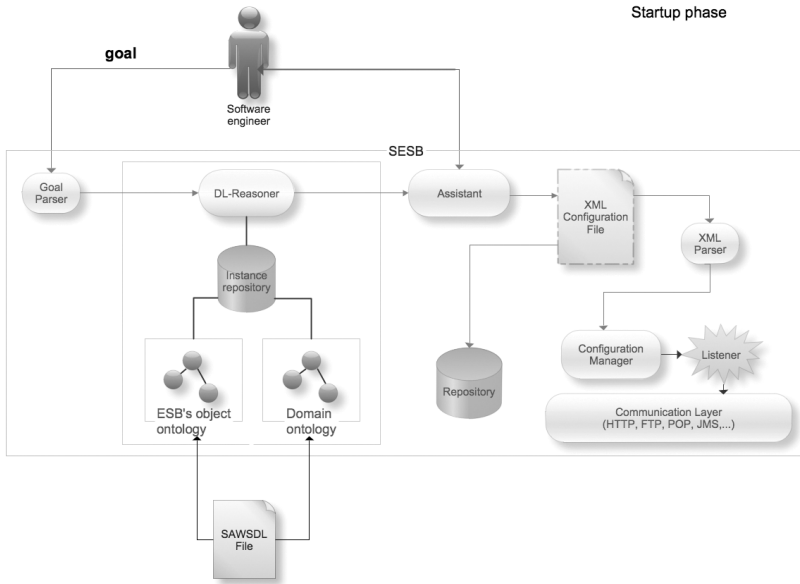


Fig. 2. SESB architecture

3.1 Building on the ESB

The first approach uses an ESB as platform to deploy different components that fulfill each process of the Semantic Web Services life cycle in an independent way. It means that there exists a component responsible for discovery, another component responsible for composition, etc. Each one of these components are deployed as different Web Services on the ESB. Therefore, the ESB acts as the mechanism that manages the communication and coordination processes among these components. Such architecture provides the user with the sensation of an existing semantic layer.

Several works have used this approach. In [9] authors present what they have called a “semantic-based dynamic service composition and adaptation” framework. In this work, they have focused on the composition and execution processes. The composer uses WSML [10] descriptions of basic services to build more complex services using a backward-chaining reasoning method. Basically, the composition algorithm relies on looking for an appropriate set of services using the precondition and postcondition descriptions until the required goal is satisfied. If the goal has been fulfilled then an ordered list of basic services is later passed to the executor, in this case the ESB.

INFRAWEBSS³ is a European IST project in the 6th FP that also exploits this approach. The main objective of INFRAWEBSS is the development of a set of tools which will help users to create, maintain and execute Semantic Web

³ <http://www.infrawebs.org/>

Services. The developed platform known as the Infraweb Integrated Framework (IIF) is WSMO [4] compliant and it relies on the Eclipse IDE.

The system generated consist of loosely coupled INFRAWEBs units, with each unit providing tools and adaptable system components. Developers will be able to use these components to analyse, design and maintain WSMO-based Semantic Web Services throughout the whole lifecycle. Each set of tools is deployed within the ESB which is also considered an Infrawebs Unit. The IIF defines two distinctive (but linked) parts: a Design Time Phase and a Runtime Phase. The Design Time Phase consists of the tools involved during the design of Semantic Web Services (using the Eclipse IDE), whereas the Runtime Phase consists of those tools involved during the execution of Semantic Web Services (using the ESB). Some of the tools may of course be involved in both phases. This approach ensures that system users (whether developers or normal end-users) deal with the minimum (but complete) set of tools required. Furthermore, it also provides an environment which is focused on the specific needs of the users.

The INFRAWEBs Integrated Framework acts as a middleware to inter-communicate components using Web Service technology, so this network of interconnected components constitutes a SOA. INFRAWEBs delegates to the ESB the management and execution of these components. The ESB will be also able to check if the components are alive or not.

The components that INFRAWEBs defines are the following:

- SWS-D (Semantic Web Service Designer): Responsible for the design of Semantic Web Service descriptions (in particular the capabilities of the Web Service descriptions), plus goals.
- SWS-C (Semantic Web Service Composer): Responsible for the static composition of existing WSMO-based Semantic Web Services.
- DSWS-R (Distributed Semantic Web Service Repository): Responsible for the persistent storage of WSMO-based descriptions and the registry and advertisement of the available services.
- SIR (Semantic Information Router): Responsible for handling the formal descriptions (based on RDF) that are submitted in the service registration process.
- OM (Organizational Memory): Represents an organizational memory and a case-based recommender tool.
- SAM (Service Access Middleware): Responsible for guiding the user applications through the steps of semantic web service usage, including service discovery, selection and execution.
- SWS-E (Semantic Web Service Executor): Responsible for executing the Semantic Web Service descriptions.
- QoS-Monitor (Quality of Service Monitor): Responsible for collecting monitor data and calculate the metric values for the Semantic Web Service being executed.
- Security components: Offer methods for ensuring and measuring trust values to the application providers. These methods are used by the end-users applications.

3.2 Extending the ESB's Capabilities

ESB draws from traditional Enterprise Application Integration (EAI) approaches functionality in that it provides integration services such as data transformation, routing of data, adapters to applications and protocol conversion. The main difference between EAI and ESB relies on the use of proprietary internal formats. The ESB is based on today's established standards such as XSLT, WSDL, BPEL4WS, etc. This fact makes application integration not specific to each provider. Another feature that makes ESB attractive to users is that it exploits configuration more than codification. There is nothing wrong with writing code, but there is plenty of code to be written elsewhere that does not have to do with interdependencies between applications and services.

In this work, our aim is to introduce semantic into the core of the ESB taking advantage of its features. The idea consists of adding semantic annotations to the different objects that the ESB can manage, such as transports, connectors or filters. Therefore, developing a component with such behaviour provides us with reasoning capabilities to the artifacts deployed on the ESB. In this way, we could use this information to facilitate several tasks in the Semantic Web Services life cycle such as the discovery and composition.

Figure 2 depicts the architecture of the proposed SESB. As we explained in Section 2, SESB will be responsible for creating the logical path among required Web Services. To achieve this, we must start from a couple of assumptions: SESB uses available information stored as instances of an OWL-DL ontology which models the objects that the SESB can understand (filters, transports, endpoints, etc.); Web Services are annotated using SAWSDL to concepts in the ESB's object ontology and concepts in a domain ontology.

The startup phase begins when the user (the software engineer) introduces the required goal. Goals represents the user's preferences and they will be introduced to the system using WSMML. Therefore the developed tool will be WSMO compliant. After that, a parser processes the goal and sends the information to the reasoner. This component relies on the ESB's object ontology and domain ontology to get information about suitable Web Services. The system generates a first version of the configuration file using the information provided by the reasoner. In this way, the user does not have to know low level details about Web Services. The SESB will be able to check the compatibility between different Web Services and ask the user for required code such as the creation of adapters to overcome the heterogeneity of inputs and outputs. The assistant is the component responsible for providing this functionality and completing the configuration file. This file can be stored in a repository for later use. When the configuration file is completed the configuration manager processes it and prepares the system to receive messages in the execution phase.

4 Related Work

The main challenge among researchers in the Semantic Web Service field lies in overcoming the technological gap between the use of syntactic technology and

semantic technology. As we can see, many R&D projects are ongoing with the aim of bringing semantics into SOA. In many cases, the goal of these projects is to build a platform enabling the use of Semantic Web Services (INFRAWEBSS⁴, WSMX⁵, IRS-III⁶).

These platforms cover the whole Semantic Web Service life cycle enabling discovery of services based on goals, composition, registry and mediation. The developed tools rely on a top-down design, i.e., they assume that users start with an ontology implementation from which Web Services will be annotated. In this way, the first step would be to specify goals, mediators and others semantic restrictions in order to build a semantic layer that Web Services will use.

Despite being the most common way to begin a new project based on Semantic Web Service technology in the future, now it is not the most suitable way to evolve current SOA applications towards a Semantic SOA. The reason for this is that with a top-down design it is not possible to take advantage of the existing Web Services. Nowadays, a bottom-up design is necessary, which allows developers to re-use the existing applications and adapt them to the Semantic Web.

At present, some researchers have noticed this necessity and have begun to provide solutions. In this sense, a first approach has been the recent recommendation of SAWSDL⁷ by W3C, after developing a mechanism to enable semantic annotation of Web Service descriptions in WSDL 2.0. SAWSDL is independent of the language used for the ontology description. This fact makes SAWSDL suitable for being combined with different approaches such as WSMO⁴ or OWL-S⁵ as we can see in the recent works in this field⁶ [3] [2].

Current effort shows that researchers have become aware of the actual technological transition in SOA. In this sense, it is difficult to know when Semantic Web Services may be used among ICT enterprises without any limitation. For the moment, researchers should postpone the development of new platforms that cover the SWS life cycle focusing their effort on obtaining a solution to overcome the current transition problem between SOA and Semantic SOA. This last issue has contributed to the development of the described work.

5 Conclusion and Future Work

In this paper, we describe how to build a Semantic Enterprise Service Bus combining several technologies such as OWL, SAWSDL and SOA. This kind of tool allows software engineers to apply a bottom-up design to deploy a solution that relying on the Semantic Web Services approach. This is an ongoing project that aims to develop a platform to overcome the problems of the current SOA, i.e. finding the most suitable service for a certain requirement among thousands of different services or building a complex service from other simple services. We

⁴ <http://www.infrawebs.org/>

⁵ <http://www.wsmx.org>

⁶ <http://kmi.open.ac.uk/projects/irs/>

⁷ <http://w3.org/TR/sawSDL>

are now focused on developing and improving some of the described components within the SESB.

As future work we plan to validate the platform using a real use case. For that, we propose the development of adapters or wrappers over existing SOA applications as a future extension of the described work. These adapters will allow the application of a semantic layer over implemented Web Services which will be reusable in the proposed SESB. In this way, we expect to implement an automatic or semi-automatic tool to annotate Web Services using SAWSDL over concepts in an ontology. For that proposal, we have even considered evaluating and reusing an existing tool known as *Radiant* [11]. This tool will be incorporated into the SESB to facilitate the deployment of non-annotated Web Services.

Acknowledgements

Supported by Grants CVI-267 (Junta de Andalucía), TIN2005-09098-C05-01 (Spanish Ministry of Education and Science), P07-TIC-02978 (Junta de Andalucía) and FIT-350503-2007-6 (Spanish Ministry of Industry, Tourism and Commerce: Plan Avanza)

References

1. Chappell, D.A.: Enterprise Service Bus. O'Reilly, Sebastopol (2004)
2. Kourttesis, D., Paraskakis, I.: Combining SAWSDL, OWL-DL and UDDI for Semantically Enhanced Web Service Discovery. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 614–628. Springer, Heidelberg (2008)
3. Paolucci, M., Wagner, M., Martin, D.: Grounding OWL-S in SAWSDL. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSOC 2007. LNCS, vol. 4749, pp. 416–421. Springer, Heidelberg (2007)
4. Roman, D., et al.: Web Service Modeling Ontology. Applied Ontology 1(1), 77–106 (2005)
5. The OWL Services Coalition. OWL-S 1.1 Release (2004)
6. Vitvar, T., Kopecky, J., Viskova, J., Fensel, D.: WSMO-Lite Annotations for Web Services. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 674–689. Springer, Heidelberg (2008)
7. Farrell, J., Lausen, H. (eds.): Semantic Annotations for WSDL and XML Schema. W3C Recommendation (August 2007)
8. McGuinness, D.L., van Harmelen, F.: OWL Web Ontology Language Overview. W3C Recommendation (February 2004)
9. Hibner, A., Zielinski, K.: Semantic-based Dynamic Service Composition and Adaptation. In: IEEE SCW 2007, pp. 213–220 (2007)
10. de Bruijn, J., Fensel, D., Lausen, H.: D34v0.1: The Web Compliance of WSML. Technical report, DERI (2007), <http://www.wsmo.org/TR/d34/v0.1/>
11. Gomadam, K., Verma, K., Brewer, D., Sheth, A.P., Miller, J.A.: A tool for semantic annotation of Web Services. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005, vol. 3729, Springer, Heidelberg (2005)

Distributed Workflows: The OpenKnowledge Experience

Paolo Besana¹, Vivek Patkar², David Glasspool¹, and Dave Robertson¹

¹ University of Edinburgh

² UCL Department of Oncology

Abstract. Software systems are becoming ever more complex, and one source of complexity lies in integrating heterogeneous subsystems. Service Oriented Architectures are part of the answer: they decouple the components of the system. However normally SOA is used from a centralised perspective: a single process invokes remote services, unaware of being part of a workflow. We claim that the centralised, or orchestration-based, approach cannot scale well with increasing complexity and heterogeneity of the components, and we propose an alternative distributed, or choreography-based, approach, that forces developers to think in terms of actors, roles and interactions. We first present the OpenKnowledge framework, designed according to choreography-based principles and then show how a complex, distributed model for managing the triple assessment of patients suspected with breast cancer can be easily implemented using this framework.

1 Introduction

Software systems are getting more and more complex. An important source of the complexity is the need to integrate different, heterogeneous subsystems. Service oriented architectures decouple the components of complex systems: every component exposes services that are accessible through the network using standard methods. Decoupling is a good software engineering practice, as it reduces the interdependencies between components and, if well designed, simplifies reusability of the services in different systems.

Complex systems can be pulled together, invoking services belonging to different and possibly external systems using workflow languages like BPEL or YAWL. These framework are based on a centralised, imperative paradigm: a central process controls everything, and the other services are passive, unaware of being part of a workflow.

We claim that this approach does not scale well with the growing complexity of systems: we advocate a different paradigm, based on the choreography of actors. We believe that the design of systems can gain from this paradigm, independent of the deployment technique. In fact, the choreography paradigm forces the developers to think in terms of the actors, their roles and their interactions in complex scenario, making them explicit.

The OpenKnowledge¹ project has allowed us to develop a fully distributed, peer-to-peer framework, focussed on shared interaction models that are executed by peers. Based on this framework, we have tested the paradigm in different scenarios of varying level of heterogeneity and complexity. In this paper we will focus on a choreography-based implementation, based on the OpenKnowledge framework, of the assessment procedure followed by a patient suspected of breast cancer. In Section 2 we explain our claims about the choreography-based architecture, at the core of the OpenKnowledge project, presented in Section 3. In Section 4 we describe the triple assessment scenario, comparing the centralised and the distributed models. Finally, in Section 5 we show how we applied the OpenKnowledge framework to the assessment scenario.

2 Choreography- and Orchestration-Based Architectures

Our claim, as stated in the introduction, is that a choreography-based architecture forces the developers to think distributed applications from a different perspective that scales better with an increasing number of interacting actors. A distributed application becomes an set of interactions between actors that take different roles. The paradigm is independent from actual implementation and deployment: the implementation can be a complete peer-to-peer, fully open architecture, or a more traditional, closed architecture where every component is certified, and deployed by certified administrator, or it can be something in between. What is different is how the application is conceived, and the type of middleware that connects the pieces.

For example, an online pharmacy that provides a service for buying prescribed drugs can be designed using an orchestration-based language such as YAWL [10] or using a choreography-based approach. Figure 1 shows a YAWL workflow for the service. It runs as a central process started by the reception of a request from a customer. The process invokes a remote service for verifying the prescription, invokes another remote service for ordering the delivery of the drug and calls another service for charging the cost (it can be the national health service, a private insurance or a direct payment system). The roles are implicit: the focus is on the flow of activities. Figure 2 shows a distributed workflow for the same application: it focuses on the actors' roles and on their interaction. The choreography-based approach forces us to analyse more explicitly the domain of the problem. In writing this simple example, I quickly found out that I needed to specify the ID of doctor to contact, and the ID of the funding body. Thinking about this, it was evident that the information was connected to the customer ID. The same information could be obviously defined in the orchestration-based architecture, but because the roles are implicit, the analysis is independent from the representation.

The choreography model, especially if implemented with an open, peer-to-peer architecture, requires designers to address issues of heterogeneity and brokering. Peers are likely to be different and they need to understand each other. The

¹ www.openk.org

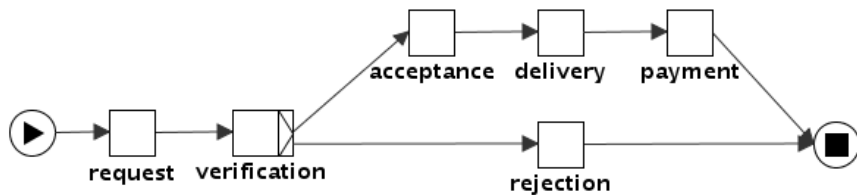


Fig. 1. Workflow for the purchase of a prescribed drug, expressed in YAWL

same services may be available from many peers, and the search and discovery process can be complex, especially if it needs to be perform at real time.

3 OpenKnowledge

The OpenKnowledge kernel [9] provides the layer that assorted services and applications can use to interact using a choreography-based architecture able to deal both with the semantic heterogeneity of the actors and with their discovery.

The framework allows a direct translation of a choreography oriented design, such as the activity diagram in Figure 2, into an executable application. The core concept is the *interaction models*, performed by different applications and service providers. These actors are the *participants* of the interactions, and they play

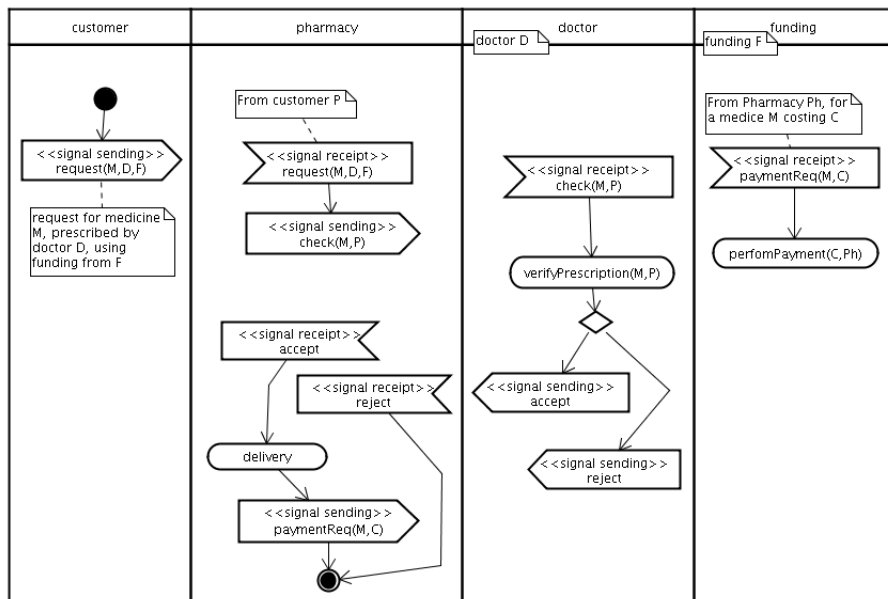


Fig. 2. Distributed workflow for the purchase of a prescribe medicine, expressed as an activity diagram

$$\begin{aligned}
& a(\textit{pharmacy}, Ph) :: \\
& \textit{request}(M, D, F) \Leftarrow a(\textit{patient}, P) \textit{ then} \\
& \textit{check}(M, P) \Rightarrow a(\textit{doctor}, D) \textit{ then} \\
& \left(\begin{array}{l}
\textit{accept} \Leftarrow a(\textit{doctor}, D) \textit{ then} \\
\textit{null} \Leftarrow \textit{getDrugCost}(M, C) \textit{ and } \textit{deliver}(M, P) \textit{ then} \\
\textit{paymentReq}(M, C) \Rightarrow a(\textit{fund}, F)
\end{array} \right) \\
& \textit{or} \\
& \textit{reject} \Leftarrow a(\textit{doctor}, D)
\end{aligned}$$

Fig. 3. LCC clause for the pharmacist role

roles in them. In an interaction all the roles have equal weight; the behaviour of all the participants and in particular their exchange of messages are specified. The roles in the interaction models are played by the participants, called *peers*.

Interaction models are written in Lightweight Coordination Calculus (LCC) [7,8] and published by the authors on the *distributed discovery service* (DDS) with a keyword-based description [5]. LCC is an executable choreography language based on process calculus. An interaction model in LCC is a set of clauses, each of which defines how a role in the interaction must be performed. Roles are described by their type and by an identifier for the individual peer undertaking that role. Participants in an interaction take their *entry-role* and follow the unfolding of the clause specified using a combinations of the sequence operator (*'then'*) or choice operator (*'or'*) to connect messages and changes of role. Messages are either outgoing to (*'=>'*) or incoming from (*'<=>'*) another participant in a given role. A participant can take, during an interaction, more roles and can recursively take the same role (for example when processing a list). Message input/output or change of role is controlled by constraints defined using the normal logical operators for conjunction and disjunction. In its definition, LCC makes no commitment to the method used to solve constraints - so different participants might operate different constraint solvers (including human intervention). Figure 3 shows the LCC clause for the pharmacy role described in the interaction of Figure 2. The clause highlights how close is the LCC transposition from the specifications represented with a UML diagram.

The peers that want to perform some task, such as buying a medicine or providing prescription verification service, search for published interaction models for the task by sending a keyword-based query to the DDS. The DDS collects the published interaction models matching the description (the keywords are extended adding synonyms to improve recall) and sends back the list.

Interaction models and peers are designed by possibly different entities, and therefore the constraints and the peers' knowledge bases are unlikely to be perfectly corresponding. The heterogeneity problem is dealt splitting the task in three phases and limiting its scope. We have already seen the first phase, performed by the DDS: the interaction descriptions are matched using a simple query expansion mechanism. Then the peers compare the constraints in the received interaction models with their own capabilities, and finally the peers need to map the terms appearing in constraints and introduced by other peers [2].

Constraint annotations

```
@annotation( @role(pharmacy), @annotation( @variable(M),
    drug(name,dose) ) )
@annotation( @role(pharmacy), @annotation( @variable(P),
    patient(name,surname,date_of_birth,address(street,post_code)) ) )
```

Java method annotation

```
@MethodSemantic(language='tag',
    params={'patient(family_name,birthday,street,post_code)',
        'medicine(name,dose)'} )
public boolean deliverMedicine(Argument P, Argument M) {...}
```

Fig. 4. Annotations for the constraint $deliver(M,D)$ and for a corresponding method

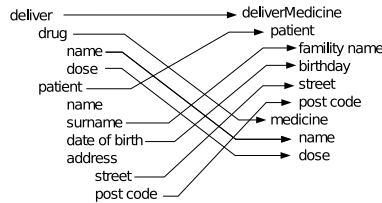


Fig. 5. Adaptor for constraint $deliver(M,P)$

The scope of the matching problem is limited to the specific interaction model in the second phase, and to the specific interaction run in the third phase.

The peer capabilities are provided by plug-in components, called OKC. An OKC exposes a set of Java methods that are compared to the constraints in the interaction models. The comparison is performed between the signatures of the constraints and of the methods, transforming them into trees and verifying their distance [3,4]. The signatures can be annotated with the semantics of each parameter, which can be structured terms, as shown in Figure 4. The comparison process creates *adaptors*, that bridge the constraints to the methods, as shown in Figure 5. An adaptor has a confidence level, that reflects the distance between the constraint and the best matching method: the average of all the confidences of constraints gives a measure of how well the peer can execute an interaction, and it is used to select the most fitting one. Once the peer has selected an interaction, it advertises its intention of interpreting one of its roles to the discovery service by subscribing to it. Figure 6 shows the state of network when roles in an interaction are subscribed by at least one peer.

When all the roles are filled, the discovery service chooses randomly a peer in the network as coordinator for the interaction, and hands over the interaction model together with the list of involved peers in order to execute it.

The coordinator first asks each peer to select the peers they want to interact with, forming a mutually compatible group of peers out of the replies and making the task of selecting the best team for a task a distributed activity. The selection is not always necessary: peers can subscribe signalling that they interact with everybody.

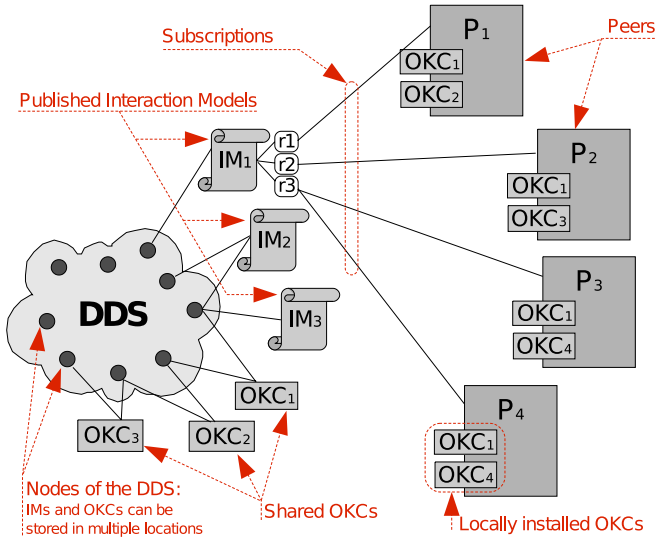


Fig. 6. OpenKnowledge architecture

While different implementations are possible, in the OpenKnowledge kernel the coordinator executes the interaction, instantiating a local proxy for each peer. The remote peers are contacted only to solve constraints in the role they have subscribed.

4 Medical Guidelines

Gaps between medical theory and clinical practice are consistently found in health service research. Care procedures can often differ significantly between different health centres, with varying outcomes for the patients, and many medical errors could be avoided if standard procedures were followed consistently. One of the causes of discrepancies in care is the difficulty in distributing and sharing efficiently the large amount of information that is continuously produced by medical research.

These issues have pushed the development of clinical practice guidelines. Several studies have shown that published guidelines can improve the quality of care. However, most such guidelines are provided as booklets, often hundreds of pages long, and covering relatively narrow fields or specific pathologies. Hundreds of guidelines are available, and generalist doctors are expected to be aware of, and follow, the guidelines relevant to each patient. The result is that guidelines are rarely followed, and inconsistencies in medical treatments are not reduced as much as hoped.

Information technology can improve the situation. Many clinical guidelines provide informal descriptions of workflows and rules that can be translated into formal representations, executable by machine, and research has suggested that

computerised clinical supports can improve practitioner compliance with the guidelines [1]. One such formal model, of a diagnosis workflow for patient suspected of breast cancer, is presented in [6].

4.1 Breast Cancer

Breast cancer is the most commonly diagnosed cancer in women, accounting for about thirty percent of all such cancers. One in nine women will develop breast cancer at some point in their lives. In the UK, women with symptoms that raise suspicion of breast cancer are referred by their GP to designated breast clinics in local hospitals. To increase the accuracy of diagnosis, a combination of clinical examination, imaging and biopsy - known together as triple assessment - is recommended for qualifying women.

The first element of triple assessment consists of gathering the patient details and clinical examinations, done by breast surgeon. If the clinical examination reveals an abnormality then the patient is referred to a radiologist for imaging which consist of either ultrasound, mammography or both. If either the examination or imaging findings warrant it then a specific type of biopsy is performed, either by a radiologist or a surgeon, and the tissue is sent to a pathologist for an examination. The collective results from all three tests influence the final management of the patient. A small number of "worried well" patients may not qualify for either imaging or biopsy and could be discharged straight away. As the entire clinical process is distributed among three different disciplines and involves a number of different clinicians, a very close co-ordination and good communication between those involved is essential for the smooth running of the clinic.

4.2 Centralised Model

The model presented in [6] is designed according to a centralised principle, and the abstract workflow is shown in Figure 7. The workflow is executed on a server², and it is focussed on the activities to performs and in their relations. As

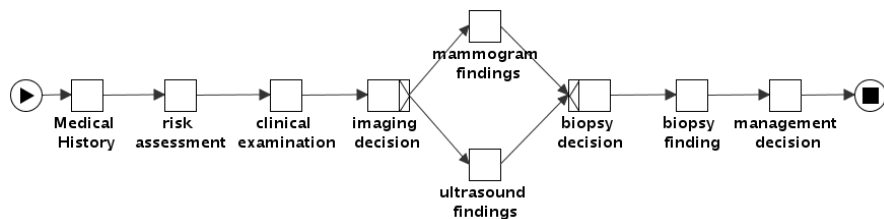


Fig. 7. Centralised representation of the triple assessment

² A demo is available at: http://www.acl.icnet.uk/lab/tallis/Sample06_Triple_Assessment_workflow.htm

described in the previous section it involves different clinicians. However this is not explicitly represented in the model: roles are enforced by requiring different permissions to access the activities. An activity queues a task to the todo list of a user, who finds it when they log in. When they finish the task, the activity is marked as completed and the workflow proceeds.

4.3 Distributed Model

The distributed nature of the triple assessment procedure is not very clear from the centralised workflow of Figure 7. A more thorough analysis is shown in the UML activity diagram of Figure 8. The four main participants: the breast surgery service (BSS), in charge of the first three activities in the workflow in Figure 7, the breast imaging service (BIS), responsible for the imaging decision and for the two alternative possible examinations, the breast pathology service (BPS), in charge of the biopsy, and the multi-disciplinary team (MDT), responsible for the final decision together with the surgery service.

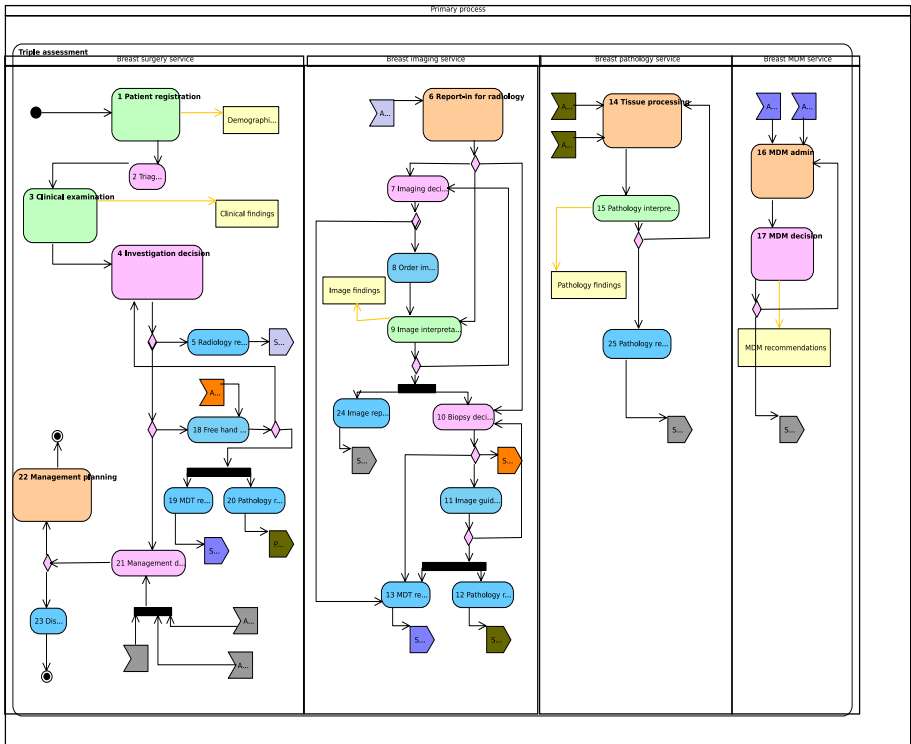


Fig. 8. Activity diagram for the triple assessment

5 Triple Assessment in OpenKnowledge

The aim of implementing the triple assessment process using the OpenKnowledge kernel was to obtain a proof of concept: verify if LCC was expressive enough for the task, and determine how the choreography approach influences the design of the system. Given the similarity between the activity diagrams as specification language and the LCC formalism, the task was rather straightforward.

The first step was the conversion from the activity diagram in Figure 8 into a LCC protocol. The work of extracting the activity diagram from the flowchart of the activity had already cleared the identity of the actors in the interaction. However, the activity diagram in the figure is the result of working on a specification intended for a centralised model and thus it includes only actors whose activities are represented in workflow. The conversion of the diagram into an LCC interaction model showed the need to include the patient as a participant.

In the test, the interaction model is published on the DDS by the peer playing the breast surgery service role, that directly subscribes to it after receiving acknowledgement of the publication. The other peers search for interaction models for triple assessment, and receive the interaction previously published by the BSS. They compare the constraints in the role they want to perform with their local OKCs, and if the match is good enough, they subscribe. In complex interactions like this a role may include many subroles, some shared with other roles: a peer needs to be able to solve the constraints in all the subroles reachable by the entry role. Figure 9 shows a fraction of the exchanged messages in a run of the interaction.

In the real world situation, the services subscribe first - and more than one service can subscribe to the same role - and the patient, or possibly her generalist doctor, subscribes to the patient role when required to go through an assessment after being identified as at risk. The other services remain always subscribed: every time a new patient subscribes, a new interaction is started. The patient is able to select the services she wants to interact with, possibly based on geographical distance or previous experiences.

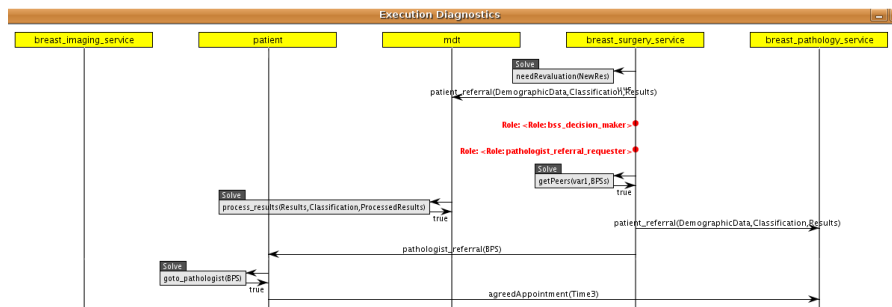


Fig. 9. Fragment of the sequence diagram of a run of the triple assessment

As we said in Section 3, the way constraints are solved is independent from the kernel: a method in an OKC can simply be a wrapper to a web service, a query to a database, a call to an external legacy application, etc. The execution of the constraints is asynchronous: the coordinator of the interaction sends a message to the peer currently active, that forwards the request to the OKC bound to the constraint. As we said earlier, the binding is found during the comparison between interactions and available OKCs.

The peers can be stand-alone applications on the practitioners' computers, that show messages to the users to alert them of an incoming patient, or wait for the users to fill up partially pre-compiled forms. Because solving constraints is decoupled from the interaction models, the peer could also be a servlet running on a web server, similarly to the current implementation of the process: the user logs in and finds the task to perform, such as a form to compile, or an update to her calendar. Similarly, the peer can be a process on a server, and the constraints can be solved sending SMS messages to a mobile phone.

6 Conclusion

In this paper we have seen how the design and implementation of complex systems, such as the application for managing patients with suspected breast cancer, can gain from a distributed, choreography-based paradigm. While the paradigm is independent of the actual implementation and deployment, OpenKnowledge provides an operational framework for quickly setting up distributed systems. In OpenKnowledge systems are composed around interaction models, that coordinate the peers' behaviours by specifying the roles they can take, the exchange of messages between the roles and the constraints of the messages. Peers participate in the interaction taking one (or more) roles: in order to participate they need to compare the constraint in the roles with their available services and subscribe to the interaction on a distributed discovery service, that initiates interactions when their roles are filled.

As shown in the medical example, interaction models map rather accurately their specification as activity diagrams, making the implementation straightforward. Moreover, discovery of interaction and semantic matching between constraints and peers services allows reusability of components.

References

1. Garg, A.X., Adhikari, N.K., McDonald, H., Rosas-Arellano, M., Devereaux, P.J., Beyene, J., Sam, J., Haynes, R.B.: Effects of computerised clinical decision support systems on practitioner performance and patient outcome: a systematic review. *JAMA* 293, 1223–1238 (2005)
2. Besana, P., Robertson, D.: How service choreography statistics reduce the ontology mapping problem. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) *ASWC 2007 and ISWC 2007*. LNCS, vol. 4825, pp. 44–57. Springer, Heidelberg (2007)

3. Giunchiglia, F., Yatskevich, M., McNeill, F.: Structure preserving semantic matching. In: Proceedings of the ISWC+ASWC International workshop on Ontology Matching (OM), Busan, KR (2007)
4. McNeill, F., Shvaiko, P., Pane, J., Giunchiglia, F., Yatskevich, M., Besana, P.: Approximate structure preserving semantic matching. In: ECAI 2008 (2008)
5. Kotoulas, S., Siebes, R.: Deliverable 2.2: Adaptive routing in structured peer-to-peer overlays. Technical report, OpenKnowledge
6. Patkar, V., Hurt, C., Steele, R., Purushotham, A., Williams, M., Thomson, R., Fox, J.: Evidence-based guidelines and decision support services: a discussion and evaluation in triple assessment of suspected breast cancer. *British Journal of Cancer* 95, 1490–1496 (2006)
7. Robertson, D.: Multi-agent coordination as distributed logic programming. In: International Conference on Logic Programming, Sant-Malo, France (2004)
8. Robertson, D., Walton, C., Barker, A., Besana, P., Chen-Burger, Y., Hassan, F., Lambert, D., Li, G., McGinnis, J., Osman, N., Bundy, A., McNeill, F., van Harmelen, F., Sierra, C., Giunchiglia, F.: Models of interaction as a grounding for peer to peer knowledge sharing. In: *Advances in Web Semantics*, vol. 1 (in press)
9. Siebes, R., Dupplaw, D., Kotoulas, S., Perreau de Pinninck, A., van Harmelen, F., Robertson, D.: The openknowledge system: an interaction-centered approach to knowledge sharing. In: Proceedings of the 15th Intl. Conference on Cooperative Information Systems, CoopIS (2007)
10. van der Aalst, W.M.P., Aldred, L., Dumas, M., ter Hofstede, A.H.M.: Design and implementation of the yawl system. In: Persson, A., Stirna, J. (eds.) *CAiSE 2004*, vol. 3084, pp. 142–159. Springer, Heidelberg (2004)

SD-Core: A Semantic Middleware Applied to Molecular Biology

Ismael Navas-Delgado, Amine Kerzazi, Othmane Chniber,
and José F. Aldana-Montes

E.T.S.I. Informática. Computer Languages and Computing Science Department,
Boulevard Louis Pasteur s/n,
29071 Málaga, Spain
{ismael,kerzazi,chniber,jfam}@lcc.uma.es

Abstract. This paper describes a middleware for building Semantic Web applications. The main idea behind this work is to help developers build Semantic Web applications by providing them with the main components for this task. This set of components has been implemented and made available using a Web Demo tool (<http://khaos.uma.es/SD-Core>). In addition, this middleware has been applied to implement two Semantic Web Tools: the Khaos Ontology-based Mediator Framework (KOMF) and the Semantic Field Tool (SemFiT). In this paper we focus on KOMF that has been used to build a Semantic Tool for integrating data from biological databases.

Keywords: Semantic Middleware, Information Integration, Life Science.

1 Introduction

Semantic Web research has been gaining popularity since the initial proposal of Tim Berners Lee. Nowadays, this research is producing novel technology that is being integrated in enterprise applications. In this context, the development of Semantic Web based applications has had to address several problems: to choose a component for dealing with ontologies, to deal with ontology relationships (usually available as ontology alignments) and to relate non-semantic resources with semantics through annotation tasks. These new issues have caused developers significant problems when estimating the real cost of applications

The essential role of middleware is to manage the complexity and heterogeneity of distributed infrastructures. On the one hand, middleware offers programming abstractions that hide some of the complexities of building a distributed application. On the other hand, a complex software infrastructure is necessary to implement these abstractions. Instead of the programmer having to deal with every aspect of a distributed application, it is the middleware that takes care of some of them.

Ontologies serve various needs in the Semantic Web, such as storage or exchange of data corresponding to an ontology, ontology-based reasoning or ontology-based navigation. Building a real Semantic Web application requires designers to combine different existing software modules. However, existing Semantic Web components are usually difficult to be located.

In this context the Semantic Web Framework [1] has a structure in which applications are described using simple components, providing a classification and analysis of existent tools, but they do not define even component interfaces. Our approach overcomes the design of a generic infrastructure by describing bigger components because the analysis of the possible Semantic Web applications indicates that some combinations of simple components are shared in all of these applications.

Another related work is KAON2 (<http://kaon2.semanticweb.org/>), which provides the following features: “an API for programmatic management of OWL-DL, SWRL, and F-Logic ontologies; a stand-alone server providing access to ontologies in a distributed manner using RMI; an inference engine for answering conjunctive queries (expressed using SPARQL syntax); a DIG interface, allowing access from tools such as Protégé; and a module for extracting ontology instances from relational databases”. However, this system is not able to deal with annotated resources, to enable a way of locating them taking their semantics into account.

We aim to develop a middleware which will hide semantics details to programmers by providing them a set of working components. As a first step towards this goal we have designed an infrastructure for developing Semantic Web applications. An infrastructure is generally a set of interconnected structural elements that provide the framework supporting an entire structure.

The goal of the proposed middleware is to provide useful components for registering and managing ontologies and their relationships, and also metadata regarding the resources committed or annotated with the registered ontologies, which means a practical step towards building applications in the Semantic Web. The main advantage of using a middleware for the development of Semantic Web applications is that software developers can reuse components, reducing the implementation costs.

This infrastructure has been instantiated as a Java implementation (SD-Core). This instantiation has been used for developing the Khaos Ontology-based Mediator Framework (KOMF), which aims to produce a way of developing mediation systems by taking advantage of existent components. KOMF has been successfully instantiated in the context of the Molecular Biology for integrating disperse data sources [2]. This implementation can be used to produce solutions requiring the access to integrated information. Thus, we have developed a tool for obtaining information about protein structures using this implementation.

Section 2 describes the proposed middleware, the Semantic Directory Core, and its instantiation for annotating data retrieving resources. Section 3 shows the design of KOMF and how it has been applied to integrate biological data sources for solving domain specific problems. Finally, conclusions are presented.

2 Semantic Directory Core

This section presents the generic middleware for the development of Semantic Web applications. The analysis of different architectural proposals and Semantic Web applications makes it clear that a Semantic Web application must have these characteristics:

- Ontologies are used to introduce semantics.
- A single and common ontology is not available for most of the domains. Ontology management and alignment is necessary.

- Resources are annotated with different ontologies, even in the same domain. In the context of SD-Core we define a resource as any software available through an URL.
- Resources need to be located by means of explicit semantics.

Summarizing the list of requirements, we can deduce that ontology and resource managers are necessary components for most of the applications. In addition, we can find relationships between ontologies and resources (this is one of the main characteristics of Semantic Web applications). These relationships are rich, and we can take advantage of them in the development of Semantic Web applications.

2.1 Infrastructure for the Middleware

The proposed infrastructure is based on a resource directory, called Semantic Directory Core, SD-Core (Figure 1). We define the SD-Core as "a set of core elements to build Semantic Web applications, and it is made available as a server to register semantics, providing services to query and browse all the registered semantics". In order to formally define the elements that the SD-Core will manage, we have defined its internal elements by means of metadata ontologies.

Semantic Directories in general are semantically described to provide a way of semantic interoperability with other Semantic Web applications. This semantics will be described in terms of two metadata ontologies, Ontology Metadata and Resource Metadata. These ontologies publicly available are internally managed through a Metadata Manager. In order to reach a useful implementation we have to define these metadata ontologies and which metadata manager we are going to use in the SD-Core.

SD-Core is composed of three interfaces (Figure 1), which tend to be the minimum set of elements for building a wide range of applications for the Semantic Web, in charge of the main tasks it provides: 1) the Ontology Metadata Repository Interface; 2) the Semantic Register Interface; and 3) the Resource Metadata Repository Interface.

The *Ontology Metadata Repository Interface* is an interface, which offers different types of access to the information of resources related with ontologies registered in the SD-Core. The basic access operation is to search ontologies as instances of the Ontology Metadata. Ontology registration implies generating a new instance of an Ontology Metadata ontology concept. Ontology Metadata should provide several kinds of concepts to describe ontologies.

The *Semantic Register Interface* is in charge of relating resources with several of the registered ontologies. When registering a resource, these interface implementations will generate an instance of the Resource Metadata containing relationships between this resource and previously registered ontologies. Thus, it will be possible to take advantage of registered resources by means of the ontologies registered in the SD-Core. Once a resource has been registered, the SD-Core monitor (the application in charge of ensuring that all components work correctly) will repetitively test if it is available. In the case that a resource is not available (reachable) it is marked as not available temporarily. Thus, if a user/application asks for resources complying with certain characteristic, un-available resources' URLs will not be returned.

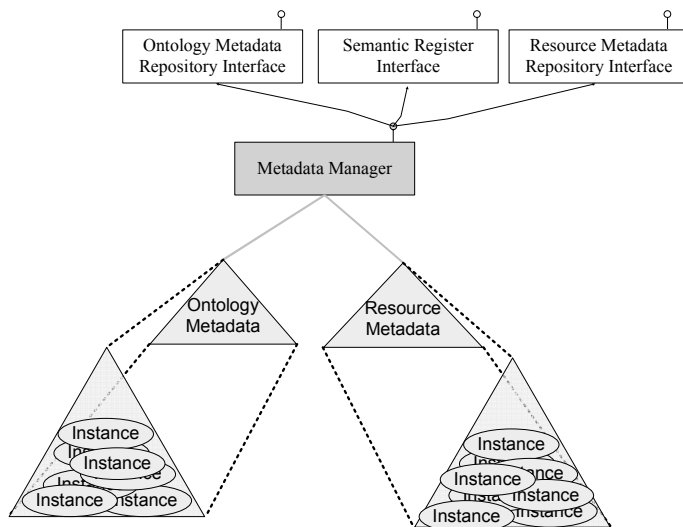


Fig. 1. SD-Core concept

The registration of a resource will generate an instance of the resource concept in the Resource Metadata. This instance will contain information about the resource. The *Resource Metadata Repository Interface* is an interface for registering and accessing information about resources, which provides methods for locating resources based on their URL, name, relationships with domain ontologies, etc. The basic access operations are to search instances of the Resource Metadata ontology.

2.2 SD-Data: The Specialization of the SD-Core to Deal with Data Providing Resources

In order to make use of the SD-Core to deal with data providing resources¹, we have specialized the proposed middleware to provide additional methods for this kind of applications, and replacing the Ontology Metadata and Resource Metadata by two specific OWL ontologies: OMV [3] and SDMO. This specialization is called the SD-Data (Figure 2).

We use OMV to register additional information about ontologies to help users locate and use them. The metadata scheme contains further elements describing various aspects related to the creation, management and use of an ontology.

SDMO is the ontology in charge of registering information about resources and relationships between these resources and ontologies registered in the SD-Data. SDMO and OMV are related by a concept included in SDMO, which provides a way of relating resources (SDMO instances) with registered ontologies (OMV instances). The current version of SDMO is composed of five concepts, OMV, Resource, Mapping, Similarity and User:

¹ In SD-Data the resources are software applications that are able to provide data as a response to queries. Specifically, we propose the registration of resources able to provide XML documents as result of queries expressed using XQuery.

- OMV: this class is used to link resources with registered ontologies (as instances of the OMV ontology). This class contains the ontology name and URL.
- Resource: this class is used to store information (query capabilities, schema, query interface, name and URI) about resources.
- Mapping: this class is used to set the relationships between resources and ontologies. Each mapping is related with a similarity instance that establishes the similarity between ontology concepts and resource elements.
- Similarity: the similarity class contains three properties (concept1, concept2 and similarity-Value) to establish the similarity between an ontology concept and a resource element.
- User: this class is added in order to deal with users in the applications.

As the two metadata ontologies are described using OWL, we have chosen Jena (<http://jena.sourceforge.net/>) as Metadata Manager in SD-Data. However, the access to the registered information it is possible to use a reasoner like DBOWL [4], that requires to be installed in the Web Server to be accessed from the SD-Data.

The use of this or another reasoner can be carried out through the DIG API (<http://dl.kr.org/dig/>). In this way, the SD-Data (and also SD-Core) can be changed for using another DIG compliant reasoner by installing and replacing DBOWL for this Reasoner.

On the other hand, SD-Data extends the SD-Core to enable additional ways of registering resources. Thus, the different methods provided enable registration at different levels: making explicit the relationships with registered ontologies or allowing the SD-Data to calculate them by means of an internal matching tool. The registration of resources will generate instances of the Mapping and Similarity classes, indicating the relationships between the resource and registered ontologies. This information could be obtained from the input parameters of the registration process, or they could be obtained automatically using a matching tool.

The resources registered are supposed to have an XMLSchema, which will be provided in the registration call. This assumption limits the type of resources that can be registered in the SD-Data. These resources have to be data providing resources, so they have a specific schema that can be directly related with one or more registered ontologies. Additional methods have been added to the Resource Metadata Repository to retrieve information related with this kind of resources:

- `getSchema`. This method locates the schema of a resource registered in the directory.
- `listMappings`. These methods list the mappings of a resource with one or more ontologies of the SD-Data. The resources are located taking into account the possible input parameters.

The SD-Data is considered a Semantic Web application because it provides basic elements for dealing with semantics without any additional component. However, more complex applications can be developed by the use of SD-Data. Thus, Semantic aware applications make use of Semantic Directories to find the semantics of resources registered in them, accessing the information through the existing resources. These resources have to be registered in Semantic Directories, but this will not involve making changes on them. On the other hand non Semantic aware applications can directly access these resources.

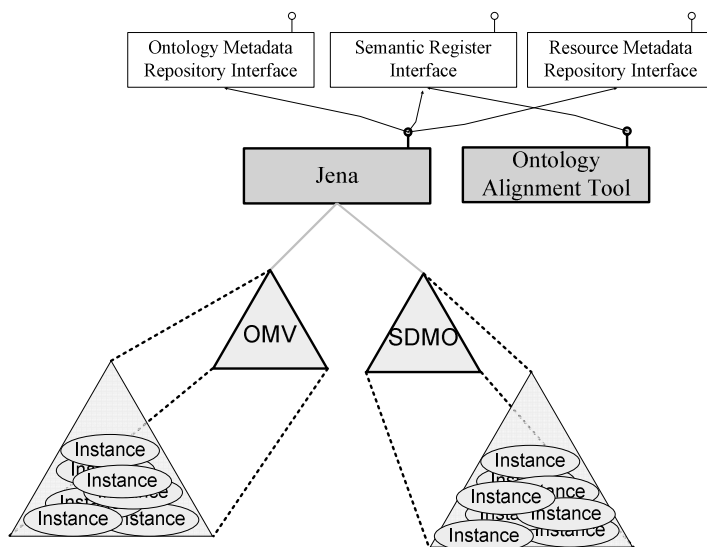


Fig. 2. SD-Data Architecture

3 KOMF: The Khaos Ontology-Based Mediator Framework

The need for data integration started when the number of applications and data repositories began to grow rapidly. The first approaches appeared in the 80's, and formed the basis for the research in this area. The evolution continued over mediator based systems, such as DISCO [5], TSIMMIS [6] and Garlic [7]. Then, agent technology was used in some systems like InfoSleuth [8] and MOMIS [9]. Finally, the new technologies appearing have been used in data integration: XML (MIX [10]) and ontologies (OBSERVER [11]). In the specific field of biological data there are the following examples: TAMBIS [12], BioDataServer [13], KIND [14], BioZoom [15], BioKleisli [16], DiscoveryLink [17] and BioBroker [18]. Studying the most relevant data integration systems have allowed us to determine the main elements of a data integration system, and so to extract the pattern for building this kind of system.

The SD-Data has been successfully applied in the development of a Semantic Web application (Figure 3): The Khaos Ontology-based Mediator Framework (KOMF). It uses SD-Data for managing semantics, and has been used to develop an end-user application, AMMO-Prot (the AMine Metabolism Ontology Protein tool) [2].

The architecture of the proposed Ontology-Based Mediator Framework is composed of three main components:

- Controller: the main task of this component is to interact with the user interface, providing solutions described in terms of one of the ontologies registered in the SD-Data.
- Query Planner: the task of this component is to find a query plan (QP) for the user query. The current planner has been implemented including the most basic reasoning mechanisms to take advantage of described semantics (subsumption

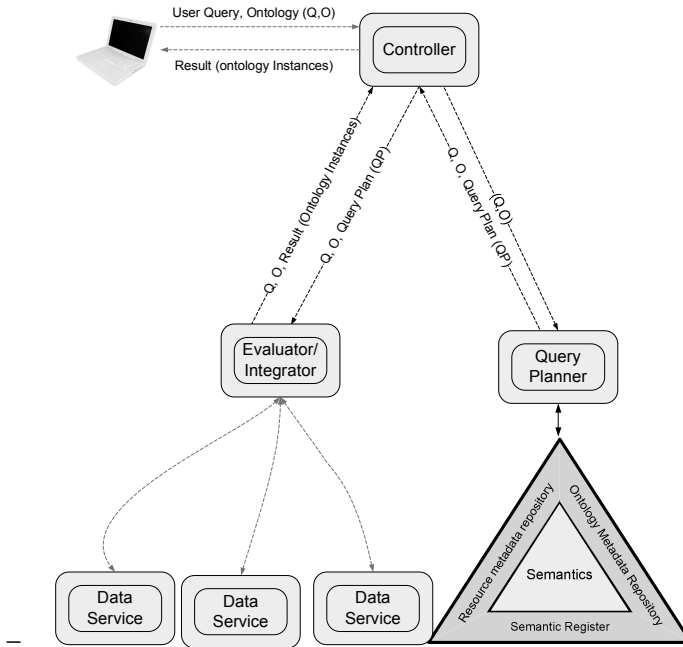


Fig. 3. KOMF architecture

and classification). Thus, if a query includes a concept this query will be expanded to include the semantic descendants. The mappings are also important in this process and are used to find if the query pattern matches one or more patterns in the mappings.

- Evaluator/Integrator: this component analyzes the query plan (QP), and performs the corresponding call to the data services involved in the sub-queries (SQ_1, \dots, SQ_n) of the query plan (R_1, \dots, R_n). This component will obtain a set of XML documents from different data services. Results from data services (R_1, \dots, R_n) are composed by this component, in this way, obtaining the results of the user query. The current implementation of this component uses the mappings to translate the XML document to ontology instances, and then a conjunctive query evaluator is applied to the set of instances found.

In our work, the sources are made available by publishing them as Web Services (named Data Services). Our primary goal here is to integrate databases accessible via internet pages. In this context, wrappers are an important part of the internal elements of data services. These services, independently of the development process, are distributed software applications that receive queries in XQuery and return XML documents.

In the context of mediator development, the process of registering resources in a SD-Data implies finding a set of mappings between one or several ontologies and the data service schema (usually expressed as an XMLSchema document). These mappings will be the key elements to integrate all the data sources, and these mapping will be the way in which the resource semantics are made explicit. The mappings used

are defined as a pair (P, Q) . P is a set of path expressions on the resource schema, and Q a query expression in terms of the ontology. In a first approach we have chosen XPath as the language to express P , and conjunctive queries to Q .

3.1 KOMF Instantiation: Example of Application on the Life Science Domain

For any field of knowledge, and particularly in Life Sciences, research on Semantic Web infrastructures and applications can be especially helpful to improve efficiency in the finding, collection and organization of data stored in the growing number of resources which make their semantics explicit.

In the context of Life Sciences, the Systems Biology framework is being merged. It is supported by all high-throughput methods which generate large amounts of data that cannot be covered simply by the human mind. This field includes a wide variety of concepts and methods but, in general, it can be considered the analysis of living systems, through studying the relationships among the elements in response to genetic or environmental perturbations, with the ultimate goal of understanding the system as a whole. A "system" can be considered at different levels, from a metabolic pathway or gene regulatory network to a cell, tissue, organism or ecosystem. The number of information repositories and services for biological elements (molecules, cells, etc) is growing exponentially. Consequently, Systems Biology is the prototype of a knowledge-intensive application domain for which the Semantic Web should be particularly relevant.

This application type has been instantiated as a fully functional tool developed in the Amine System Project (ASP, <http://asp.uma.es/WebMediator>) using the SD-Data for registering semantics taking advantage of an OWL ontology developed in this project. Thus, KOMF has been applied to solve a data integration problem in this biological domain. The main advantage of this mediator is that data services can be easily plugged into the system. The most complex parts need to be developed by Semantic Web, data integration and domain experts: the wrapper building and its publication as a data service and the description of the mappings between the data service schema and the domain ontology used. However, we believe that wrapper building can be semi-automated to facilitate the development of this kind of applications. Besides, the search of mappings can be performed by using automatic tools, and then these mappings can be reviewed by domain experts.

The proposed solution has been applied to solve a specific problem for biologist [2]: *"A common and useful strategy to determine the 3D structure of a protein, which cannot be obtained by its crystallization, is to apply comparative modelling techniques. These techniques start working with the primary sequence of the target protein to finally predict its 3D structure by comparing the target polypeptide to those of solved homologous proteins"*.

Thus, [2] describes a tool for solving this problem by integrating different databases through KOMF. However, other applications can be built using this approach. In this way, we are working in a tool for integrating information from metabolic databases to extract knowledge about the behavior of living cells.

4 Conclusion

This paper describes a middleware for enabling the development of semantic Web tools. This middleware, SD-Core, is based in an architecture designed to manage metadata about ontologies and annotated resources. SD-Core has been extended to deal with data providing resources, producing the SD-Data, which has been applied to develop a framework for integrating data sources. This framework, KOMF, provides the main components of a mediation system. These components has been implemented and used to integrate biological information, showing the viability of the proposal to provide solutions for real life applications.

Acknowledgments. Supported by Grants CVI-267 a (Junta de Andalucía), TIN2005-09098-C05-01 (Spanish Ministry of Education and Science), and the Junta de Andalucía project P07-TIC-02978.

References

1. Leger, A., Gómez-Pérez, A., Maynard, D., Zyskowski, D., Hartmann, J., Euzenat, J., Dzbor, M., Zaremba, M., del Carmen Suárez-Figueroa, M., García-Castro, R., Palma, R., Dasiopoulou, S., Costache, S., Vitvar, T.: Architecture of the semantic web framework. Technical report (February 2007)
2. Navas-Delgado, I., Montañez, R., Pino-Ángeles, A., Moya-García, A.A., Urdiales, J.L., Sánchez-Jiménez, F., Aldana-Montes, J.F.: AMMO-Prot: ASP Model Finder. BMC Bioinformatics. Biomed. Central Ltd. (2008) ISSN: 1471-2105 (JCR Impact factor: 3.617)
3. Hartmann, J., Sure, Y., Haase, P., Palma, R., del Carmen Suarez-Figueroa, M.: Omv ontology metadata vocabulary. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005, vol. 3729, Springer, Heidelberg (2005)
4. del Mar Roldan-Garcia, M., Aldana-Montes, J.F.: DBOWL: Towards a Scalable and Persistent OWL Reasoner. In: ICIW, pp. 174–179 (2008)
5. Tomasic, A., Amouroux, R., Bonnet, P., Kapitskaia, O., Naacke, H., Raschid, L.: The distributed information search component (disco) and the world wide web. In: Proceeding of the 1997ACM SIGMOD International Conference on Management of Data, pp. 546–548 (1997)
6. Garcia-Molina, H., Papakonstantinou, Y., Quass, D., Rajaraman, A., Sagiv, Y., Ullman, J., Vassalos, V., Widom, J.: The tsimmis aproach to mediation: Data models and languages. *Journal of Intelligent Information Systems* 8(2), 117–132 (1997)
7. Haas, L., Kossmann, D., Wimmers, E., Yang, J.: An optimizer for heterogeneous systems with nonstandard data and search capabilities. *Data Engineering Bulletin* 19, 37–44 (1996)
8. Ksiezzyk, T., Martin, G., Jia, Q.: Infosleuth: Agent-based system for data integration and analysis. In: Proceedings of the 25th International Computer Software and Applications Conference on Invigorating Software Development, p. 474 (2001)
9. Beneventano, D., Bergamaschi, S., Castano, S., Corni, A., Guidetti, R., Malvezzi, G., Melchiori, M., Vincini, M.: Information integration: The momis project demonstration. In: Proceedings of the 26th Intrnational Conference on Very Large Data Bases, pp. 611–614. Morgan Kaufmann Publishers, San Francisco (2000)
10. BornhÅovd, C., Buchmann, a.A.P.: A prototype for metadata-based integration of internet sources. In: Jarke, M., Oberweis, A. (eds.) CAiSE 1999, vol. 1626, p. 439. Springer, Heidelberg (1999)

11. Mena, E., Kashyap, V., Sheth, A.P., Illarramendi, A.: OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. In: Conference on Cooperative Information Systems, pp. 14–25 (1996)
12. Stevens, R., Baker, P., Bechhofer, S., Ng, G., Jacoby, A., Paton, N.W., Goble, C.A., Brass, A.: TAMBIS: Transparent access to multiple bioinformatics information sources. *Bioinformatics* 16, 184–186 (2000)
13. Lange, M., Freier, A., Scholz, U., Stephanik, A.: A computational support for access to integrated molecular biology data. In: German Conference on Bioinformatics (2001), <http://www.bioinfo.de/isb/gcb01/poster/lange.html#img-1>
14. Gupta, A., Ludascher, B., Martone, M.E.: Knowledge-based integration of neuroscience data sources. In: Proceedings of the 12th International Conference on Scientific and Statistical Database Management (SSDBM), Berlin, Germany, July 2000, IEEE Computer Society, Los Alamitos (2000)
15. Liu, L., Buttler, D., Critchlow, T., Han, W., Pâques, H., Pu, C., Rocco, D.: BioZoom: Exploiting source-capability information for integrated access to multiple bioinformatics data sources. In: Proceedings of the 3rd IEEE Symposium on Bioinformatics and BioEngineering (BIBE 2003), Washington, DC, March 10–12, IEEE Computer Society Press, Los Alamitos (2003)
16. Davidson, S., Overton, C., Tannen, V., BioKleisli, W.L.: A digital library for biomedical researchers. *International Journal of Digital Libraries* 1, 36–53 (1997)
17. IBM Corp: DiscoveryLink (August 23, 2004), <http://www.ibm.com/discoverylink>
18. Aldana, J.F., Roldán-Castro, M., Navas-Delgado, I., Roldán-García, M.M., Hidalgo-Conde, M., Trelles, O.: Bio-Broker: a tool for integration of biological data sources and data analysis tools. *Software: Practice and Experience* 36(14), 1585–1604 (2006)

Towards Knowledge in the Cloud

Davide Cerri¹, Emanuele Della Valle^{1,2}, David De Francisco Marcos³,
Fausto Giunchiglia⁴, Dalit Naor⁵, Lyndon Nixon⁶, Kia Teymourian⁶,
Philipp Obermeier⁶, Dietrich Rebholz-Schuhmann⁷, Reto Kruppenacher⁸,
and Elena Simperl⁸

¹ CEFRIEL – Politecnico of Milano, Via Fucini 2, 20133 Milano, Italy
{davide.cerri,emanuele.dellavalle}@cefriel.it

² Dip. di Elettronica e Informazione, Politecnico di Milano, Milano, Italy
emanuele.dellavalle@polimi.it

³ Telefonica Investigacion y Desarrollo, Valladolid, Spain
davidfr@tid.es

⁴ Dipartimento Ingegneria e Scienza dell'Informazione, University of Trento, Povo,
Trento, Italy
fausto@disi.unitn.it

⁵ IBM Haifa Research Laboratory, Haifa, Israel
DALIT@il.ibm.com

⁶ Institute of Computer Science, Free University of Berlin, Berlin, Germany
nixon@inf.fu-berlin.de

⁷ European Molecular Biology Laboratory, European Bioinformatics Institute,
Heidelberg, Germany
rebholz@ebi.ac.uk

⁸ Semantic Technology Institute, University of Innsbruck, Austria
{reto.kruppenacher,elena.simperl}@sti2.at

Abstract. Knowledge in the form of semantic data is becoming more and more ubiquitous, and the need for scalable, dynamic systems to support collaborative work with such distributed, heterogeneous knowledge arises. We extend the “data in the cloud” approach that is emerging today to “knowledge in the cloud”, with support for handling semantic information, organizing and finding it efficiently and providing reasoning and quality support. Both the life sciences and emergency response fields are identified as strong potential beneficiaries of having “knowledge in the cloud”.

1 Introduction

Knowledge in the form of semantic data¹ is becoming increasingly ubiquitous in the Internet, but important steps towards scalable, dynamic systems to support collaborative work with distributed, heterogeneous knowledge are still missing. Following the *data in the cloud* paradigm that is emerging today (such as Amazon S3²), in this paper we propose a future vision of “knowledge in the cloud”.

¹ Promoted by the W3C Semantic Web activity <http://www.w3.org/2001/sw>

² <http://aws.amazon.com/s3>

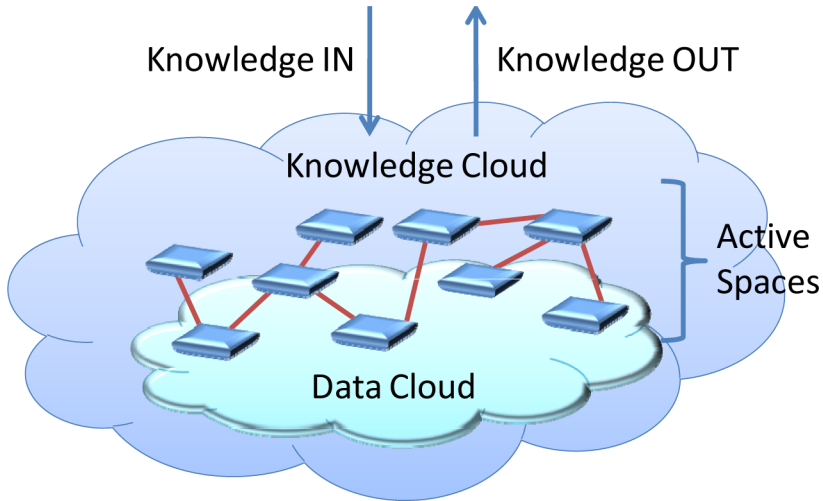


Fig. 1. The vision of knowledge in the cloud

This vision is applicable to critical collaborative tasks as different as life sciences and emergency response. In the life sciences, large scale knowledge about protein functions must be optimally organized and ubiquitously available so that new inferences can be made and passed as input to collaborating analysis activities. In emergency response, deriving facts from knowledge about flooding must be available to emergency workers in a time critical manner in order that they collaborate effectively.

Data in the cloud refers to the cloud storage idea, where data is stored somewhere on the Web through abstract APIs with loose schemas and without *any* constraint of space, availability and scalability. Clients can *completely* rely on the data cloud and count on loose coupling, as access is not tied to particular access patterns dependent on the use of specific schemas. This loose coupling is similar to the one provided by Triplespace Computing [12], an emerging coordination paradigm combining semantics, tuplespaces [3] and Web Service technology for the persistent publication of knowledge and the coordination of services using that knowledge.

We believe in the possibility to merge the Triplespace Computing paradigm with data in the cloud forming the “knowledge in the cloud” vision, which incorporates support for knowledge (semantic data), coordination (collaboration) and self-organization (internal optimisation). Our view on “knowledge in the cloud” is illustrated in Figure 1.

Distributed applications are increasingly sharing their data in the cloud. To take advantage of the “knowledge in the cloud” vision, firstly the semantic knowledge must be extracted from the underlying data. This knowledge is shared in the overlying knowledge cloud (which provides ubiquitous access) in active (i.e. with reasoning) spaces (which provide collaboration and coordination).

In the rest of the paper, we present how we believe this vision can be realized using the approaches mentioned above (Section 2). In Section 3, we provide a short description of two scenarios that would benefit from applying "knowledge in the cloud". Finally in Section 4 and 5, we briefly compare our vision with other on-going work and we discuss its potential impacts.

2 Approach

In this section we introduce the approaches to be taken and applied to realise the "knowledge in the cloud" vision outlined in this paper.

2.1 Cloud Storage

Cloud Storage provides support for always available, ubiquitous access to hosted data. A virtualized entity of data is made available online and hosted on a variety of multiple virtual servers, mostly hosted by third parties, rather than being hosted on dedicated servers. Such cloud storage approaches are being increasingly used in real world scenarios to make application access to data independent from a physical machine or connection.

While such solutions exist for Internet-scale, highly available data storage and management, they do not address specifically the case where that data is in fact knowledge, i.e. semantic data expressed by a formal logical model. This includes the map-reduce paradigm from Google [4] and the basic S3 cloud storage service of Amazon. Although it provides a highly available storage service that is cheap to maintain, it mainly targets large data objects that are not frequently modified and is not context aware.

To support "knowledge in the cloud", a special cloud based storage service is required which is tailored for handling semantic data, which includes allowing execution of reasoning over the data and can support distributed query processing.

2.2 Self-organizing Systems

Self-organizing systems seek to implement a decentralised, autonomous organization of data within a widely distributed system. This is able to provide stability, dynamicism and scalability. We consider these important requirements on "knowledge in the cloud", where large amounts of knowledge must be always-available in the cloud.

Swarm intelligence can be an innovative solution for "knowledge in the cloud" [5]. Swarm individuals are the active entities that are able to observe their neighborhood, to move in the environment, and to change the state of the environment in which they are located. For example, they might move between nodes in a distributed system, picking up and dropping data in their path, performing calculations or executing new processes. Despite the lack of central control, such systems demonstrate emergent global behaviour patterns.

SwarmLinda [6] is to our knowledge the only system which combines tuple-space computing and self-organization principles. SwarmLinda uses the ant colony pattern, to cluster tuples according to their content, and efficiently answer queries through following scents left by tuple-carrying ants. Ants can also load-balance tuples on nodes. The principles of SwarmLinda can be used to optimize tuple distribution and retrieval in large-scale space-based systems. First implementation results [7] demonstrate the expected clustering of data in a decentralised manner, leading to greater efficiency, dynamicism and scalability in the system.

Knowledge in the cloud must go beyond the state of the art, extending swarming to support the distribution and retrieval of semantic data, creating a new research field of semantic swarms. Now notions of fitness, relatedness, similarity etc. must be inferred from semantic data. By that, we aim to achieve intelligent swarm intelligence. This fundamental shift will result in new and powerful decentralized and scalable algorithms for applying swarm computing to triplespaces.

2.3 Distributed Querying and Reasoning

Distribution of a system is a necessity for achieving Internet scalability, hence distributed query processing is a requirement on "knowledge in the cloud". Research in distributed semantic query processing started recently and predominantly leans on optimization techniques for general database query languages.

There are known query processing optimizations in distributed databases as well as rule-based rewriting techniques for queries which deliver optimization on a logical level independent from the physical implementation. Current work in distributed semantic querying includes DARQ [8] and SemWIQ [9].

Swarm-based approaches have the potential to perform emergent problem solving through divide and conquer [10], however they have not yet been applied to query optimisation in distributed semantic systems, as we require for "knowledge in the cloud". The aim here is to move reasoning out of the reasoner on top of the data store, and into the cloud of self-organizing semantic data, exploring new forms of distributed reasoning where inference processes co-operating in the cloud can optimize the reasoning task for large scale knowledge clouds.

Reasoning individuals as part of the swarm can have a rather limited set of reasoning capabilities compared to a full blown theorem prover - thus it is called micro-reasoning. We divide reasoning tasks across these individuals and apply swarm algorithms to them, such that the collective reasoning in the swarm provides inference on data expressed by common semantic models (subsets of Description Logics and Logic Programming) and therefore permits the solution of complex problems. Given the reduction of the required schema information for each individual, and the distribution of this schema information in any case, we expect that such micro-reasoners can provide more efficient and scalable reasoning services for well defined reasoning sub-tasks.

A further extension of reasoning is to support "active spaces", which allow rules to be stored in tuples besides RDF and OWL semantic information, triggering automatic generation of knowledge beyond the capabilities of the RDF

and OWL languages and supporting notifications to clients of new knowledge states which have been inferred from the existing knowledge.

2.4 Trust and Data Quality

The "knowledge in the cloud" can be distributed with a Peer-to-Peer (P2P) architecture. P2P brings advantages for data and knowledge management systems such as efficient routing, fault tolerance, and low cost of ownership. At the same time, it opens up new research challenges such as those related to the assessment of the quality of data. In critical collaboration scenarios as foreseen in this paper, the quality of the semantic data is vital for correct inferences from the knowledge.

The notion of data quality is well understood in the context of centralized information systems. Quality assessment in these approaches relies on the assumptions that data can be accessed and evaluated centrally, and that any piece of data in the system can be retrieved upon a user request. These data quality metrics cannot be directly applied in the P2P setting due to its decentralized, dynamic and subjective nature. In fact, in a P2P system, query results are often incomplete. Furthermore, it is hard to evaluate the correctness of a query result due to the fact that each peer maintains its subjective view on the data. Thus, the same data can be considered as correct by one peer and as incorrect by another. See [11] for a detailed discussion.

Thus, the standard data quality metrics have to be reconsidered in order to reflect the peculiar characteristics of P2P systems. However, very little has been done so far in this direction. One such case is the work done as part of the project OpenKnowledge (EU IST-27253) [12]. Here, data quality is measured by combining trust values and the result of a matching process [13,14,15] that aligns interaction models of different peers [16].

In "knowledge in the cloud", we must go beyond the state-of-the-art because we need to develop, implement, and evaluate a methodology for assessing the quality of answers within the knowledge cloud, a knowledge-centric self-organizing P2P system. Many spaces in the system, owned by different and unrelated entities, may be suitable for the fulfilment of a request. In this case, the requester must decide which space(s) to choose; in doing this, different information can be taken into account. A space in the system, seen as a resource, can include in its description also a quality value, which is self-declared: the owner of that resource autonomously states the quality of what he provides. A requester, after having used a resource, can evaluate the quality of that resource, and in case also give a feedback that states his opinion about the quality of that resource.

To the best of our knowledge, this methodology would be the first attempt to provide a basis for qualitative and quantitative evaluation of query answers within this kind of P2P system.

3 Application Scenarios

We identify two application scenarios that can benefit from "knowledge in the cloud" technology, and can provide valuable use cases: life science and emergency response.

3.1 Life Science

The life science community already connects a large number of scientists sharing data sets and collects results of analyses in large scale collaborative tasks. However, as these data sets become increasingly semantic (enabling new forms of inference of results) and forms of collaboration more complex, the large scale data infrastructure must support expressive knowledge models which can scale and coordinate large numbers of interdependent processes over the infrastructure. We foresee "knowledge in the cloud" as fulfilling this requirement.

The proposed scenario delivers facts from the scientific literature and from the bioinformatics scientific data resources into the "knowledge in the cloud" infrastructure. Appropriate fact databases (e.g., UniProtKb [17]) and associated ontologies are both available, i.e. their content can be expressed as semantic data and thus can be fed into the knowledge cloud to make data ubiquitously available to the community, including the automatically inferred new knowledge.

In general terms, two basic research goals form the core of ongoing research in the life science community: (1) the identification of novel biological principles that explain the functioning of cells and organisms (biomedical science), and (2) the identification of agents (e.g., chemical entities, modifications to genes) that can be used to improve the functioning of cells or organisms, in particular under the condition that they malfunction. "Knowledge in the cloud" provides the required infrastructure to carry out this research.

3.2 Emergency Response

Emergency monitoring and coordination activities usually involve a range of different organizations and teams at various administrative levels with their own systems and services. In a real emergency situation, these teams need to maximally coordinate their efforts in order to work together on time-critical tasks. However, because these teams come from different organizations, they generally have incomplete or even, contradictory knowledge of the actual emergency situation. Therefore, the coordination of numerous heterogeneous actors, policies, procedures, data standards and systems results in problems in collaborative work with respect to data and knowledge analysis, information delivery and resource management, all of which are critical elements of emergency response management. Hence "knowledge in the cloud", by being semantically rich, coordinated and scalable, can be very valuable in supporting such scenarios.

Emergency situations involve multiple user profiles, where each of them performs a different task. The emergency situation in our target use case, flooding, (see [16] for a description of the scenario implementation in the OpenKnowledge

project) involves about 10 different user profiles such as emergency coordinators, firefighter coordinators, police officer coordinators, medical staff, bus/ambulance drivers, and others. The platform to be developed is intended to find practical applications in municipal emergency services, where it can be used for: (i) finding points of possible failures and bottlenecks in emergency activities through simulating emergency situations; (ii) training of emergency personnel through running a simulator, thus improving their level of preparedness in real emergency situations; and, (iii) for supporting decision making procedures at runtime in a real emergency situation, in which taking an important decision quickly is an indispensable asset.

4 Related Works

The paradigm of data or computing in the cloud as an alternative to fully integrated approaches are becoming more and more prominent. Companies like Amazon with its Simple Storage Service S3³ and the Elastic Compute Cloud EC2⁴ or GigaSpaces Technologies Ltd. with its space-based computing platform XAP [18] explore the concept of cloud computing for the realization of scalable and fault-tolerant applications. Recently GigaSpaces released a cloud application server that enables their platform on top of the EC2 framework. The two companies argue that this combination enables an on-demand model of cloud computing with a usage-based pricing model. Moreover, it provides cloud portability, on-demand scalability of the cloud, self-healing, and high-performance. In contrast to our proposal however, neither Amazon nor GigaSpace yet exploit the benefits of semantic technologies.

Table 1. Related research efforts

Project	Key Area	Relevant Technology	Application
TripCom	Triplespace Computing	Triplespaces	Basis for active spaces
Open Knowledge	Coordination semantics and models in P2P networks	Interaction models, matching of lightweight ontologies, trust and service composition via reputation	Basis for trust and data quality
LarKC	Massive distributed reasoning	Reasoning algorithms that relax completeness and correctness requirements	Incorporation in active spaces and distributed querying
Reservoir	Cloud computing	Cloud storage	Basis for cloud platform

³ <http://aws.amazon.com/s3>

⁴ <http://aws.amazon.com/ec2>

Loosely coupled solutions to storage integration were also recognized as important by the database community. [19] propose an integration framework that does not rely on a priori semantic integration, but rather on the co-existence of data sources. Such a system – called dataspace – delivers a layer of core functionalities on top of the actual data providers that only virtually exposes a data integration platform to applications and users. This allows applications to focus on their own functionality rather than on the challenges of data integrity and efficiency of integration. Furthermore, by keeping the individual data sources on distinctive machines conserves the advantages of distributed systems: no centralized server (thus avoiding bottlenecks with respect to performance and data access), robustness, and scalability of data volumes and the number of users.

While these projects from industry and academia show again the relevance of enhanced cloud computing solutions, our vision adds another dimension by integrating semantic technologies and further results from various ongoing and past European research efforts. In Table II, we show some selected research projects related to our 'Knowledge in the Cloud' vision. First and foremost, the TripCom (www.tripcom.org) project lays the foundations with its Triplespace platform. OpenKnowledge (www.openk.org) is a project that provides similar interaction machinery as required for 'Knowledge in the Cloud' based on a quite different approach. The project LarKC (www.larkc.eu) deals with massive reasoning over distributed knowledge in the Semantic Web, especially via relaxation of traditional inference requirements on completeness and correctness. The project Reservoir investigates cloud storage, and its results could serve as basis for our data cloud.

5 Conclusions

The realization of "knowledge in the cloud" would provide valuable benefits to several scientific and industrial communities. In this paper, we described briefly the vision of "knowledge in the cloud", outlined the approaches needed to implement it, and introduced two scenarios in which it enables the necessary collaboration of large scale and distributed knowledge. Now we consider its potential wider impact.

From the scientific point of view, the "knowledge in the cloud" vision can apply self-organization algorithms, based on swarm intelligence principles, as well as Semantic Web standards to the existing cloud storage state of the art. The semantic tuplespace community can benefit from the very scalable and ubiquitous storage infrastructure of cloud storage systems. Semantic repositories can also take advantage of the distributed reasoning capabilities envisioned. The aforementioned synergies can burst into new research lines that could be worth for these communities to explore.

From the industry impact point of view, the "knowledge in the cloud" vision goes beyond and completes currently emerging trends in the enterprise, such as cloud computing (placing data in the cloud), collaboration (wikis, blogs) and knowledge management. These trends are the main pillars on which current tech-

nology aggregators develop and promote solutions to emerging business models, which demand transparent, scalable, reliable and flexible knowledge infrastructures. The impact of such an infrastructure would also be beneficial for the final users, who could enjoy greater service ubiquity, with an eased and improved interaction with systems due to the transparent usage of knowledge instead of data.

This paper has presented the research directions for the realization of "knowledge in the cloud" as well as outlined two concrete illustrations of its potential value in concrete applications. We will continue to promote the vision and push the research areas mentioned here, towards a realization of "knowledge in the cloud" as the natural next progression in and integration of semantic, coordination and cloud computing systems.

Acknowledgments

Giona McNeill, Paolo Besana, Dave Robertson, Maurizio Marchese, Juan Pane, Lorenzo Vaccari, Veronica Rizzi, Pavak Shvaiko from the Open Knowledge team are thanked for their valuable contribution. Ilya Zaijrayeu is also thanked for his feedback on these ideas.

References

1. Fensel, D.: Triple-Space Computing: Semantic Web Services Based on Persistent Publication of Information. In: IFIP Int'l. Conf. on Intelligence in Communication Systems, pp. 43–53 (2004)
2. Simperl, E., Krummenacher, R., Nixon, L.: A Coordination Model for Triplespace Computing. In: 9th Int'l. Conference on Coordination Models and Languages, pp. 1–18 (2007)
3. Gelernter, D.: Generative communication in linda. *ACM Trans. Program. Lang. Syst.* 7(1), 80–112 (1985)
4. Lämmel, R.: Google's MapReduce Programming Model – Revisited. *Science of Computer Programming* 70(1), 1–30 (2008)
5. Hinchey, M., Sterritt, R., Rouff, C.: Swarms and Swarm Intelligence. *IEEE Computer* 40(4), 111–113 (2007)
6. Charles, A., Menezes, R., Tolksdorf, R.: On The Implementation of SwarmLinda. In: 42nd Annual Southeast Regional Conference, pp. 297–298 (2004)
7. Graff, D.: Implementation and Evaluation of a SwarmLinda System. Technical Report TR-B-08-06, Free University of Berlin (2008)
8. Quilitz, B., Leser, U.: Querying Distributed RDF Data Sources with SPARQL. In: 5th European Semantic Web Conference, pp. 524–538 (2008)
9. Langegger, A., Wöß, W., Blöchl, M.: A Semantic Web Middleware for Virtual Data Integration on the Web. In: 5th European Semantic Web Conference, pp. 493–507 (2008)
10. Palmer, D., Kirschenbaum, M., Shifflet, J., Seiter, L.: Swarm Reasoning. In: IEEE Swarm Intelligence Symposium, pp. 294–301 (2005)
11. Giunchiglia, F., Zaijrayeu, I.: Making Peer Databases Interact - A Vision for an Architecture Supporting Data Coordination. In: 6th Int'l. Workshop on Cooperative Information Agents, pp. 18–35 (2002)

12. Robertson, D., Barker, A., Besana, P., Bundy, A., Chen-burger, Y.H., Giunchiglia, F., van Harmelen, F., Hassan, F., Kotoulas, S., Lambert, D., Li, G., McGinnis, J., Osman, F.M.N., Sierra, C., Walton, C.: Worldwide Intelligent Systems. In: *Advances in Web Semantics*. IOS Press, Amsterdam (1993)
13. Giunchiglia, F., Yatskevich, M., Shvaiko, P.: Semantic Matching: Algorithms and Implementation. *Journal on Data Semantics* 9, 1–38 (2007)
14. Giunchiglia, F., Yatskevich, M., Giunchiglia, E.: Efficient Semantic Matching. In: *2nd European Semantic Web Conference*, pp. 272–289 (2005)
15. Giunchiglia, F., Yatskevich, M.: Element Level Semantic Matching. In: *ISWC Workshop on Meaning Coordination and Negotiation* (2004)
16. Giunchiglia, F., McNeill, F., Yatskevich, M., Pane, J., Besana, P., Shvaiko, P.: Approximate Structure Preserving Semantic Matching. In: *7th Int'l. Conference on Ontologies, DataBases, and Applications of Semantics* (2008)
17. UniProt-Consortium: UniProt: The Universal Protein Knowledgebase. *Nucleic Acids Research* 32, 115–119 (2004)
18. Shalom, N.: The Scalability Revolution: From Dead End to Open Road - An SBA Concept Paper. *GigaSpaces Technologie* (2007)
19. Franklin, M., Halevy, A., Maier, D.: From Databases to Dataspaces: a New Abstraction for Information Management. *ACM SIGMOD Record* 34(4), 27–33 (2005)

SWWS 2008 PC Co-chairs' Message

The Web has now been in existence for quite some time, and it has produced a major shift in our thinking on the nature and scope of information processing. It is rapidly moving towards an application deployment and knowledge deployment that requires complex interactions and properly structured underlying semantics. There has been a sudden upsurge of research activity in the problems associated with adding semantics to the Web. This work on semantics will involve data, knowledge, and process semantics. The International IFIP Workshop on Semantic Web Web Semantics (SWWS 2008), which is in its fourth year, provides a forum for presenting original, unpublished research results and innovative ideas related to this voluminous quantity of research. This year a total of 18 papers were submitted to SWWS. Each of these submissions was rigorously reviewed by at least two experts. The papers were judged according to their originality, significance to theory and practice, readability and relevance to workshop topics. This resulted in the selection of nine regular papers for presentation at the workshop. This year we also included 2 invited papers by leading experts on the Semantic Web. Papers for the SWWS workshop mainly focus on the areas of ontology development, ontology management, ontology evolution, semantic interoperability, multiagent systems and biomedical ontologies. We feel that SWWS 2008 papers will inspire further research in the areas of Semantic Web and its applications. We would like to express our deepest appreciation to the authors of the submitted papers and would like to thank all the workshop attendees. We would also like to thank the program committee members and external reviewers for their efforts in maintaining the quality of papers and turning the SWWS 2008 workshop into a success. Thank you and we hope you enjoyed participating in SWWS 2008.

November 2008

Tharam S. Dillon

A Deductive Approach for Resource Interoperability and Well-Defined Workflows

Nadia Yacoubi Ayadi^{1,2}, Zoé Lacroix^{1,3}, and Maria-Esther Vidal⁴

¹ Scientific Data Management Lab, Arizona State University, Tempe AZ 85281-5706, USA

² RIADI-GDL Lab, National School of Computer Science
Campus Universitaire de la Manouba, 2010 Tunisia
{nadia.yacoubi, zoe.lacroix}@asu.edu

³ Pharmaceutical Genomics Division, Translational Genomics Research Institute
13400 E Shea Blvd, Scottsdale, AZ 85259, USA

⁴ Computer Science Department, Universidad Simón Bolívar, Caracas, Venezuela
mvidal@ldc.usb.ve

Abstract. We present a model that supports Web service description, composition, and workflow design. The model is composed of an *ontological layer* that represents domain concepts and their relationships, and a *resource layer* that describes the resources in terms of a domain ontology. With our approach the design of a workflow is expressed as a conceptual query network while its implementation is computed deductively from the selected resource descriptions. We illustrate our approach with an application case from the domain of bioinformatics.

Keywords: Ontology; Semantic Web; Integration; Mediation; Bioinformatics; Web services; Workflow; Scientific protocol; Service composition.

1 Introduction

Life sciences are continuously evolving generating more data organized in publicly available data sources whose number and size have increased exponentially for the last few years. Indeed, the 2008 update of the molecular biology databases collection includes 1078 databases [10], that is 110 more than the previous one [9] and a significant increase from 209 in the first collection published in [4]. The offering of bioinformatics tools and services follows a similar progression [3]. Thanks to this wealth, scientific experiments rely more on various digital tasks such as data retrieval from public data sources and data analysis with bioinformatics tools or services organized in complex workflows [19]. Hence, the challenges met by the scientists are to identify the resources suitable to each scientific task and compose them into executable workflows.

Scientific workflow systems such as Taverna [16] provide limited support for those two challenging tasks and scientists typically design their workflow implementation manually before entering them in a system for execution. Additionally, scientific workflows are complex queries whose implementation requires the

integration of heterogeneous resources. In this paper, we present an original deductive approach that supports resource discovery and reasoning on workflows.

Semantic information may facilitate the processes of discovering and composing resources by capturing the aim of each workflow task on one side and classifying resources on the other. This approach is used by SemanticMap [21] where bioinformatics resources are mapped to the conceptual relationships they implement in an ontology, and by ProtocolDB [11] used by scientists to express their protocol design and implementations against a domain ontology and bioinformatics resources respectively. However, neither Semantic Map nor ProtocolDB address the problem of composition of services into an executable workflow. The deductive approach presented in this paper complements ProtocolDB and SemanticMap. While, ProtocolDB allows scientists to express abstract workflows, SemanticMap identifies the resources suitable for a conceptual task *regardless of the data format of their input and output*. Implementations generated by SemanticMap do not always describe an executable workflow as resources are selected semantically rather than syntactically. Our approach is compatible with ProtocolDB and SemanticMap, but additionally, we are able to use the semantics of the services defined in terms of properties of the input and output parameters to support Web service composition. Although motivated by the field of bioinformatics, the approach presented in this paper is generic and addresses the problem of workflow implementations with resources guided by some domain knowledge.

The main contributions of the paper are:

1. A semantic model extending [13,14] that provides scientists a *semantic map* of Web services semantically annotated with respect to a domain ontology and maps a syntactic description of service inputs and outputs to corresponding conceptual collections.
2. An axiomatic system that defines the properties of the mappings between workflow design and ontologies. Mappings enable workflow implementations in terms of Web services and support resource discovery and reasoning on workflows.

The paper is organized as follows. Section 2 presents the approach and the semantic model. Workflow design and implementation are defined in Section 3. Section 4 is devoted to a case study in bioinformatics. Related approaches are discussed in Section 5. Finally, we conclude and discuss future work in Section 6.

2 The Semantic Model

We extend the model proposed in [13,14] to support semantic service composition. The proposed approach is comprised of two levels: the *ontological level* and the *resource level*. At the ontological level, an *ontology graph* captures the semantics and properties of the concepts that characterize a domain. An instance of a top-level domain ontology is depicted in Figure 1. It is composed of named nodes (e.g., **Gene**) and named directed edges (e.g., `codes_for` from the node **Gene** to the node **Protein**).

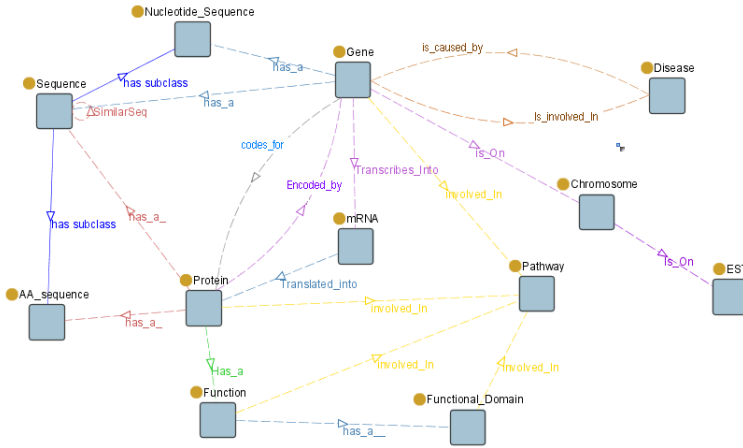


Fig. 1. A fragment of a Domain Ontology

In many domains such as biology, different ontologies have been developed to annotate data existing in disparate resources, e.g., Gene Ontology¹ [7] and Protein Ontology². In our approach, a top-level ontology graph serves as a support to organize different object-specific ontologies. Each object-specific ontology can be mapped to a node of the top-level ontology graph. For example, the Protein Ontology may be mapped to the node `Protein`.

Definition 1 (Ontology Graph). Let C be a set of class names $C = \{c_1, c_2, \dots, c_n\}$ and AN a set of association names. An ontology graph is a pair $OG = (C, LC)$, where $LC \subseteq C \times AN \times C$ is a set of labeled binary associations between the classes of C .

Definition 2 (Ontology Hierarchy). Let C be a set of class names $C = \{c_1, c_2, \dots, c_n\}$ and $OG = (C, LC)$ an ontology graph. The subgraph O_H of OG limited to the `has_subclass` relation constitutes the ontology hierarchy. A class c_1 is a subclass of c_2 (noted $c_1 \preceq_H c_2$) iff $(c_1, c_2) \in O_H$.

The relationships among ontology classes are implemented by resources of the service graph. For example, *NCBI protein* data source implements the relationships `codes_for` between `protein` and `gene`. At the resource layer, we assume that we have a set of services organized in a *service graph*. First, we define services. The input (resp., output) of services are typically defined as a list of typed variables or with a particular data format for some application domain (e.g., FASTA for DNA sequences in bioinformatics). We use collections of typed variables to provide a unified definition of bioinformatics services as follows.

Definition 3 (Basic Type). Let C be a set of class names $C = \{c_1, c_2, \dots, c_n\}$. The set of basic types τ of variables is defined by $\tau := c$, where $c \in C$.

¹ Available at <http://www.geneontology.org>

² Available at <http://proteinontology.info/about.htm>

We distinguish two types of variables: basic types which represent variables of conceptual classes and complex types which represent collections thereof.

Definition 4 (Type). *The set of types τ is defined inductively as follows:*

- $\tau := c$, where $c \in C = \{c_1, c_2, \dots, c_n\}$,
- $\tau := \text{bag}(\tau')$, τ' is a type,
- $\tau := \text{seq}(\tau')$, τ' is a type,
- $\forall n, \tau := \text{record}(\tau_1, \dots, \tau_n)$, where each τ_i is a type.

The subtyping relationship combined with the ontology hierarchy relationship on conceptual classes provide the mechanisms for reasoning.

Definition 5 (subtyping). *Let τ_1 and τ_2 be two types, τ_1 is a subtype of τ_2 (noted $\tau_1 \preceq_{\tau} \tau_2$), iff:*

- if $\tau_1 := c_1$, $\tau_2 := c_2$, and c_1 is subclass of c_2 then $\tau_1 \preceq_{\tau} \tau_2$.
- if $\tau_1 := \text{bag}(c_1)$, $\tau_2 := \text{bag}(c_2)$, and c_1 is subclass of c_2 then $\tau_1 \preceq_{\tau} \tau_2$.
- if $\tau_1 := \text{seq}(c_1)$, $\tau_2 := \text{seq}(c_2)$, and c_1 is subclass of c_2 then $\tau_1 \preceq_{\tau} \tau_2$.
- if $\tau_1 := \text{record}(c_{11}, c_{12}, \dots, c_{1p})$, $\tau_2 := \text{record}(c_{21}, c_{22}, \dots, c_{2p})$, and for each i c_{1i} is equal or subclass of c_{2i} then $\tau_1 \preceq_{\tau} \tau_2$.

We adopt the RDFS vocabulary to declare and define Web services and their semantics. A Web service is declared as an RDF triple as follows (s , `rdf:type`, `isService`). Input and output parameters of available services are declared with the RDF triple (I , s , O).

Definition 6 (Service). *Let RN be a set of resource names, V a set of typed variables. A service s_i is represented by an RDF triple (I , n_i , O) where $n_i \in RN$ is the service name and $I, O \in V$ are the input and output of s_i , respectively.*

The *service graph* models the interoperability of resources. Each data retrieval access to a data source, each link between two data sources, each application or tool is represented as a service (I , n_i , O). When the output of a service s is of same type as the input of service s' then the two services can be composed and are linked in the service graph (Figure 2).

Definition 7 (Service Graph). *Let S be a set of services $\{s_1, s_2, \dots, s_n\}$, a service graph is a pair $SG = (S, \preceq)$, where $\forall s = (I, n, O)$ and $s' = (I', n', O')$, $s \preceq s'$ iff $O \preceq_{\tau} I'$.*

The service graph denotes all possible combinations of services of S . Figure 2 depicts a service graph composed of a set of bioinformatic Web services and their interconnexions. For example, the Protein Data Bank (PDB) provides a variety of tools and resources for studying the structures of proteins. Given a protein ID, PDB allows scientists to study structure proteins and their biological functions. Because of SwissProt and PDB have compatible output and input, they are inter-linked in the service graph.

The service graph is mapped to the ontology graph and the Web service description framework is specified by a set rules as follows. A subset of these rules is defined in Table 1.

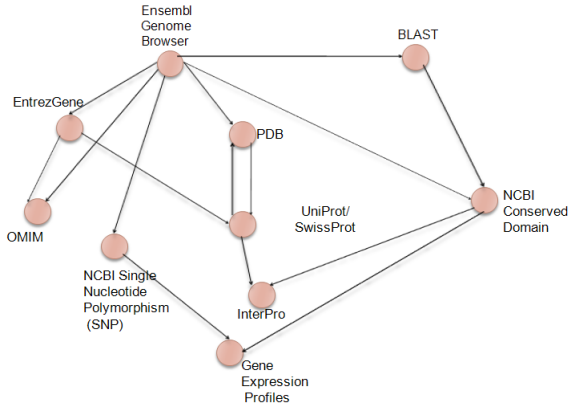


Fig. 2. An example of a service graph

Definition 8. A semantic framework for service description is 4-tuple $(OG, SG, m_{OS}, m_{OLS})$ where:

- $OG = (C, L_C)$ is an ontology graph;
- $SG = (S, \preceq)$ is a service graph;
- $m_{OS}: V \rightarrow Col(C)$ maps the input and output of services to logical concepts in C ;
- $m_{OLS}: S \rightarrow L_C$ map each service s of S to the paths in L_C it implements.

3 Well-Defined Semantic Workflows

Workflows (or equivalently protocols, pipelines, or dataflows) are complex processes composed of a *design workflow* W_D that captures its aim with respect to a domain ontology, and one or more *implementation workflows* W_I that specify the resources selected to implement each task. A workflow is defined as a generic network that is mapped to the ontology and service graphs, respectively. The mapping to the ontology graph defines the design workflow W_D , whereas the second mapping defines the implementation workflow W_I .

3.1 Workflow Network

A workflow network can be defined inductively from tasks, or well-defined basic workflow networks. Each task is defined by its task name, input type, and output type.

Definition 9 (Task). Let T be a set of task names, V be a set of typed variables. A task is a triple (i, t, o) where $i, o \in V$ and $t \in T$.

Workflow networks are composed of tasks connected by two operators: successor operator denoted by \odot and parallel operator denoted by \otimes .

Definition 10 (Workflow). Let T be a set of task names, V be a set of typed variables. The set of workflow networks W is defined recursively as follows and specified in Table 1.

- if $w = (i, t, o)$ where t is a task name then $w \in W$, $In(w)=i$, and $Out(w)=o$;
- if $w = w_1 \odot w_2$ where $w_1, w_2 \in W$ and $Out(w_1) \preceq_{\tau} In(w_2)$, then $w \in W$, $In(w)=In(w_1)$, and $Out(w)=Out(w_2)$;
- if $w = w_1 \otimes w_2$ where $w_1, w_2 \in W$, then $w \in W$, $In(w)=In(w_1) \times In(w_2)$, and $Out(w)= Out(w_1) \times Out(w_2)$.

Table 1. A subset of the axiomatic system

$\frac{[(I, s, O), (s, \text{rdf:type}, \text{isService})]}{[(s, \text{hasInput}, I), (s, \text{hasOutput}, O)]}$
$\frac{[(I, s, O), (I, \text{rdf:type}, c_1), (s, \text{rdf:type}, \text{isService})]}{[(s, \text{hasInputType}, c_1)]}$
$\frac{[(t, \text{rdf:type}, \text{isWorkflowTask})]}{[(t, \text{rdf:type}, \text{isWorkflow})]}$
$\frac{[(i_1, w_1, o_1), (w_1, \text{rdf:type}, \text{isWorkflow}), (i_2, w_2, o_2), (w_2, \text{rdf:type}, \text{isWorkflow})]}{[(i_1 \times i_2, w_1 \otimes w_2, o_1 \times o_2), (w_1 \otimes w_2, \text{rdf:type}, \text{isWorkflow})]}$
$\frac{[(i_1, w_1, o_1), (w_1, \text{rdf:type}, \text{isWorkflow}), (i_2, w_2, o_2), (w_2, \text{rdf:type}, \text{isWorkflow}), (o_1, \text{IsSubtypeOf}, i_2)]}{[(i_1, w_1 \odot w_2, o_2), (w_1 \odot w_2, \text{rdf:type}, \text{isWorkflow})]}$

The mapping of the workflow network to the ontology graph defines the design workflow and establishes that each design task is mapped to a relationship or to a path between two concept classes of the ontology graph. For example, a task that given a gene symbol returns a transcript may be mapped to the **Transcribes Into** relationship between classes **Gene** and **mRNA**.

Definition 11 (Workflow design). Let w be a workflow network, T_w the set of tasks in w , and O_G an ontology graph. A workflow design is the mapping $m_{T_w, O}: T_w \rightarrow O_G$ that defines each scientific task $(i, t, o) \in T_w$ in terms of nodes and edges in O_G .

An implementation workflow is expressed in terms of the services of the service graph that implement the concepts and paths of the design workflow.

Definition 12 (Workflow implementation). Let w be a workflow network, T_w the set of tasks in w , and S_G a service graph. A workflow implementation is a mapping $m_{T_w, S}: T_w \rightarrow S_G$ that defines each scientific task $(i, t, o) \in T_w$ in terms of nodes and edges of S_G .

Knowledge encoded by the axiomatic system previously described is combined with the knowledge that describes the workflows. The resulting knowledge base

is used to infer new properties of the available resources and workflows. This deductive system framework provides the basics to support the tasks of resource discovery, service composition and workflow reasoning required to solve the workflow interoperability problem. Resource discovery and service composition addresses the following problem. Given a workflow design return suitable workflow implementations. Suitable implementations must validate the semantics of the workflow expressed by its design and syntactic interoperability. The latter is captured by the ability to produce, when necessary, a connector between two services of compatible formats. The latter compatibility is expressed by the compatibility of types and the subtype relationship \preceq_{τ} . Knowledge represented in this framework can be enriched with QoS parameters to rank workflow implementations with respect to various criteria such as efficiency, reliability, etc. [5] Workflow reasoning extends resource discovery by allowing workflow re-use to implement partially or totally new workflows. In addition, the axiomatic system supports workflow querying by using various levels of similarities as two workflows may be similar because they have similar designs or because they have similar implementations. Both similarities are expressed by rules in the system. The whole infrastructure is implemented with deductive databases as explained in Section 3.2.

3.2 Implementation of the Approach

We implement both ontologies and Web services as two deductive databases: Deductive Ontology Base (DOB) [18] and Deductive Web Service (DWS) [1]. Each deductive database is composed of an Extensional Base (EB) and an Intensional Base (IB). The knowledge explicitly described in the ontology that defines the types of arguments of the services, is represented in the EB. On the other hand, implicit knowledge and properties that can be inferred from the axioms, are represented as intensional predicates in the IB. Meta-level predicates are provided in both bases (EB and IB) to describe explicit and implicit knowledge. For example, the description of classes, services, properties, parameters, and hierarchies are modeled in the EB. Predicates as `isClass`, `subclassOf`, `isService` are used to express explicitly that something is a class or a service and that the relationship subclass relates two classes. Domain dependent knowledge is also expressed in the EB and the IB. Domain dependent knowledge illustrates integrity constraints (implicit relationships) between modelled domain concepts. Domain dependent IB predicates explicit relationships between domain concepts in a form of rules. For example, the following IB predicate illustrates when a gene expresses a function, that is a *gene is transcribed into a set of transcripts (mRNA), each mRNA is translated into a protein which has a biological function*:

```
Expresses(g,f) ← TranscribesInto(g,mr),
                TranslatesInto(mr,p),
                Hasa(p,f).
```

From the implementation point of view, a workflow design query is defined as a deductive database query. The query is evaluated against the deductive

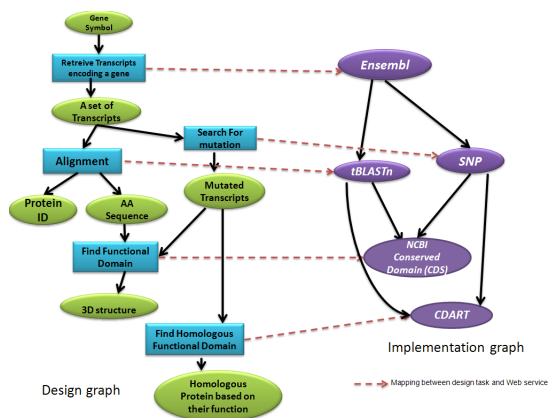


Fig. 3. Mappings between Workflow design and Workflow implementation

database that encodes all the semantics represented at the ontology and service levels, and their mappings expressed by the axiomatic system. Thus, rewriting query algorithms are to be considered in our future work. We are studying the complexity issues related to the hybrid reasoning that we are performing both at the meta-level and domain-dependant levels.

4 A Case Study in Bioinformatics

We illustrate how the approach supports the expression of scientific protocols and leads to the identification of resources and their composition into an executable workflow with the following scientific problem. Because the *human APP gene* is implicated in *Alzheimer's disease*, a scientist may be interested to know which genes are *functionally homologous* to APP. Such a question is in fact a *workflow design* as it only expresses domain knowledge and does not specify how the scientist expects to get the result. A scientific question corresponds to several paths in a domain ontology. Studies recorded in [5] reveal that biologists consider different types of relationships between scientific objects; some are precomputed and explicitly represented and stored as annotations, while others need to be calculated on-the-fly using some bioinformatics tools. The former are typically represented as edges in the ontology (e.g., the relationship *CodesFor* from *Gene* to *Protein*) whereas the latter are captured by paths combining several edges of the ontology. Each task of the workflow is mapped to a relationship between two concepts which represent the input and output parameters of the task. Each task of the workflow design illustrated in the left side of Figure 4, is mapped to a service path on the implementation graph. Here, each designed task is mapped to a single service. The following conjunctive rule expresses the workflow design semantics:

$\text{map}(\text{Workflow_1}, \text{Expresses}) \leftarrow \text{hasRDFType}(\text{Workflow_1}, \text{isWorkflow}),$
 $\text{hasInputType}(\text{Workflow_1}, \text{GeneID}),$
 $\text{hasOutputType}(\text{Workflow_1}, \text{function}).$

In our approach, we make also explicit domain-dependant knowledge that expresses constraints of the domain. For instance, *if a gene is encoded by a protein*, this means that *the nucleotide sequence of the gene codes for the amino acid sequence of the corresponding protein*.

$\text{EncodedBy}(a, n) \leftarrow \text{NuclSeq}(n),$
 $\text{CodesFor}(n, a),$
 $\text{AASeq}(a).$

We assume that we have a set of available Web services described by the set of axioms depicted in the following table.

$\frac{[(I_1, \text{Ensembl}, O_1), (I_1, \text{rdf:type}, \text{Gene_ID}), (O_1, \text{rdf:type}, \text{Transcript})]}{[(\text{Ensembl}, \text{map}, \text{TranscribesInto})]}$
$\frac{[(I_2, \text{tblastn}, O_2), (I_2, \text{rdf:type}, \text{DNA_sequence}), (O_2, \text{rdf:type}, \text{Protein_ID})]}{[(\text{tblastn}, \text{map}, \text{Translatesinto})]}$
$\frac{[(I_3, \text{CDS}, O_3), (I_3, \text{rdf:type}, \text{Protein_ID}), (O_2, \text{rdf:type}, \text{Protein_function})]}{[(\text{CDS}, \text{map}, \text{hasa})]}$

Thus, the following corresponds to a composition Web service plans.

$\frac{[(I_1, \text{Ensembl}, O_1), (I_2, \text{tblastn}, O_2), (O_1, \text{rdf:type}, \text{Transcript}), (I_2, \text{rdf:type}, \text{DNA_seq}), (\text{Transcript}, \text{IsSubtypeOf}, \text{DNA_seq})]}{[(\text{Ensembl}, \text{successor}, \text{tblastn})]}$
$\frac{[(I_2, \text{tblastn}, O_2), (I_3, \text{CDS}, O_3), (O_2, \text{rdf:type}, \text{Protein_ID}), (I_3, \text{rdf:type}, \text{Protein_ID}), (\text{Protein_ID}, \text{IsSubtypeOf}, \text{Protein_ID})]}{[(\text{tblastn}, \text{successor}, \text{CDS})]}$

The example shows how the proposed approach enables (1) service discovery, by mapping Web services to ontological relationships; (2) service composition by inferring possible Web service composition plans based on available semantic annotations; and (3) resolving mismatches by inferring compatible parameters.

5 Related Work

Semantic Web service. (SWS) technology aims at providing richer semantic specifications of Web services, in order to enable the flexible automation of service processes. The field includes substantial bodies of work, such as the efforts around OWL for Services (OWL-S) [15], the Web Service Modeling Ontology (WSMO) [17], and SAWSDL [12]. OWL-S is an OWL ontology to describe Web services, while, WSMO provides also an ontology and a conceptual framework to describe Web services and related aspects. SAWSDL is a set of extensions for WSDL, which provides a standard description format for Web services. Our approach relies on a canonical set of semantic description for Web services that we consider sufficient to describe service semantics and to perform our reasoning

tasks. We also satisfy SAWSDL standard by annotating and mapping WSDL service descriptions (inputs, outputs, and operations) with their semantic meaning in available distributed ontologies. However, our approach intends to leverage SAWSDL to support automatic composition of services based on a reasoning framework [1].

Web service in Life Sciences. One way of formalising and executing in silico experiments is to pipe together inputs and outputs of consecutive Web services in a workflow environment; this is the approach used by several existing projects in the life sciences [6,20]. First, the BioMoby [8] project offers a semantically enhanced bioinformatic registry. Web services are registered with respect to a service type ontology to classify Web services tasks and a datatype ontology to specify semantic types of input/output parameters. Because of a lack of authority to control the submission of new ontological classes used either to declare input and output parameters and to classify Web services, BioMoby ontologies show multiple inconsistencies and redundancies. In addition, only services that have been registered with compatible datatypes can be composed. Finally, [2] proposes an approach that support Web service composition task by showing how semantic information can be inferred from the mappings existing within implemented workflows.

6 Conclusion and Future work

In this paper, we propose a model that encodes the semantics required to generate Web service composition plans given a workflow design. The axioms that define the properties of the model structures are provided; they can be used by Web service composers to improve the quality of their answers. We implement the approach as a deductive database, which is comprised of an extensional database EB and an Intensional database IB. The Extensional Database encodes all the information explicitly defined in the ontologies and services, while, the Intensional Database represents all the information that can be inferred by using the proposed axioms. We present an example that illustrates the use of proposed model. The system under development will enhance ProtocolDB by identifying composition paths that lead to executable workflows. Finally, query evaluation techniques that combine meta-level and domain dependent knowledge will be defined.

Acknowledgments

The authors wish to thank Nejla Stambouli and Christophe Legendre for helping designing a biologically sound example and Maliha Aziz for discussions on formats for bioinformatics services. This research was partially supported by the

National Science Foundation³ (grants IIS 0431174, IIS 0551444, IIS 0612273, and IIS 0820457).

References

1. Ayadi, N.Y., Lacroix, Z., Vidal, M.-E., Ruckhaus, E.: Deductive web services: An ontology-driven approach for service interoperability in life science. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2007, Part II. LNCS, vol. 4806, pp. 1338–1347. Springer, Heidelberg (2007)
2. Belhajjame, K., Embury, S.M., Paton, N.W., Stevens, R., Goble, C.A.: Automatic annotation of web services based on workflow definitions. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 116–129. Springer, Heidelberg (2006)
3. Benson, G.: Editorial. *Nucleic Acids Research*, 35(Web-Server-Issue): 1 (2007)
4. Burks, C.: Molecular biology database list. *Nucleic Acids Research* 27(1), 1–9 (1999)
5. Cohen-Boulakia, S., Davidson, S., Foidevaux, C., Lacroix, Z., Vidal, M.-E.: Path-based systems to guide scientists in the maze of biological data sources. *Journal of Bioinformatics and Computational Biology* 4(5), 1069–1095 (2006)
6. Conery, J.S., Catchen, J., Lynch, M.: Rule-based workflow management for bioinformatics. *VLDB Journal* 14(3), 318–329 (2005)
7. G. O. Consortium. Gene Ontology: Tool for the unification of biology. *Nature Genetics* 25, 25–29 (May 2000)
8. T. B. Consortium. Interoperability with moby 1.0—it’s better than sharing your toothbrush! Briefings in Bioinformatics, pages bbn003+ (January 2008)
9. Galperin, M.Y.: The molecular biology database collection: 2007 update. *Nucleic Acids Res.* 35(Database issue) (January 2007)
10. Galperin, M.Y.: The molecular biology database collection: 2008 update. *Nucleic Acids Res.* 36(Database issue), D2–D4 (January 2008)
11. Kinsky, M., Lacroix, Z., Legendre, C., Wlodarczyk, P., Ayadi, N.Y.: Protocoldb: Storing scientific protocols with a domain ontology. In: Weske, M., Hacid, M.-S., Godart, C. (eds.) WISE Workshops 2007. LNCS, vol. 4832, pp. 17–28. Springer, Heidelberg (2007)
12. Kopecký, J., Vitvar, T., Bournez, C., Farrell, J.: SawSDL: Semantic annotations for WSDL and XML schema. *IEEE Internet Computing* 11(6), 60–67 (2007)
13. Kwasnikowska, N., Chen, Y., Lacroix, Z.: Modeling and storing scientific protocols. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4277, pp. 730–739. Springer, Heidelberg (2006)
14. Lacroix, Z., Raschid, L., Vidal, M.E.: Semantic model to integrate biological resources. In: Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDEW), Washington, DC, USA, p. 63. IEEE Computer Society, Los Alamitos (2006)
15. Martin, D., Burstein, M., Mcdermott, D., Mcilraith, S., Paolucci, M., Sycara, K., McGuinness, D.L., Sirin, E., Srinivasan, N.: Bringing semantics to web services with owl-s, Hingham, MA, USA, vol. 10, pp. 243–277. Kluwer Academic Publishers, Dordrecht (2007)

³ Any opinion, finding, and conclusion or recommendation expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

16. Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M.R., Wipat, A., Li, P.: Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* 20(17), 3045–3054 (2004)
17. Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., Fensel, D.: Web service modeling ontology. *Applied Ontology* 1(1), 77–106 (2005)
18. Ruckhaus, E., Vidal, M.-E., Ruiz, E.: Query evaluation and optimization in the semantic web. In: *Int. Workshop on ALPSW 2006*. CEUR-WS, vol. 287 (2006)
19. Stevens, R., Goble, C.A., Baker, P.G., Brass, A.: A classification of tasks in bioinformatics. *Bioinformatics* 17(1), 180–188 (2001)
20. Thakkar, S., Ambite, L., Knoblock, A.: Composing, optimizing, and executing plans for bioinformatics web services. *VLDB Journal* 14(3), 330–353 (2005)
21. Tufféry, P., Lacroix, Z., Ménager, H.: Semantic map of services for structural bioinformatics. In: *Proceedings of 18th International Conference on Scientific and Statistical Database Management*, Vienna, Austria, pp. 217–224 (2006)

Ontology Robustness in Evolution

Paolo Ceravolo, Ernesto Damiani, and Marcello Leida

Università degli Studi di Milano - Dipartimento di Tecnologie dell'Informazione
Via Bramante, 65 - 26013 Crema - Italy
{ceravolo,damiani}@dti.unimi.it

Abstract. This paper introduces the notion of *Ontology Robustness in Evolution* and discusses a solution based on the distinction among a stable component and a contingent component of the ontology. The stable component represents the annotation used to store data into the ontology, while the contingent component contains assertions generated by constraining the assertions in the stable component. This distinction can be used to understand which annotations can be migrated from one old version of the ontology to a new one. This way it is minimized the number of inconsistencies introduced in annotations when evolving an ontology.

Keywords: Ontology Construction, Ontology Evolution, Migration of Instances.

1 Introduction

Ontologies were originally designed as a tool for supporting Artificial Intelligence. Then engineers adopted them as an extension of shared vocabularies, in order to improve the consistency of coordination in a design process. For these historical reasons, ontologies are mainly explained and interpreted as conceptualizations of a domain [3]. But this vision can confuse because, when to manage the knowledge of an informative system, ontologies act both as a conceptualization and as a tool for data storage.

Understanding this point and formally defining a way to separate these two roles can be an important contribution in making ontologies robust to evolution. If we follow the wide tradition that see ontologies as key enablers for the Semantic Web [1] (and we adopt the OWL-DL language as a base standard [19]), the problem is how to maintain the knowledge base of annotations (Abox) consistent to the evolution of the ontology (Tbox). The solution goes toward the adoption of a policy that precisely defines the predicates used to annotate instances. This way the predicates that are functional to the annotation process provides a base to represent data stored, while the rest of the ontology provides the conceptualization of the domain.

By far, the most critical step in the implementation of an ontology-based system is modeling the knowledge base. This activity requires expertise in both modeling a domain through the use of logical languages and the specific domain one wants to model (maybe covering a wide range of different technical domains).

Different methodologies were developed in order to support ontology modeling [2]. But in general the benefit of applying a methodology impacts mainly on the quality of the model produced rather than on cost reduction. The high cost of building a knowledge-base is in contradiction with the trivial evidence that every domain evolves and the knowledge-bases related to it have to evolve as well. During a domain evolution, a dis-alignment may result between the actual domain's state and the knowledge-base. This problem is even more crucial if we consider a dynamic environment, such as for instance business modeling [16] or knowledge management. The evolution of models in dynamic environment is characterized by discontinuities because the model should ensure to consistently follow the changing dynamics of the environment. This problem is well known in the knowledge management community [10] and different aspects were addressed, such as for instance knowledge alignment, merging, versioning etc. The literature in the field is wide and composite (stressing on different scenarios, technologies, and methods). Therefore, in this paper we restrict our discussion to ontology evolution. In particular, we propose a novel approach that is based on the notion of *Ontology Robustness in Evolution*. Especially, we discuss a solution based on the distinction between two tasks covered by the ontology: *conceptualization* and *storage*.

The discussion is structured as follows: in Section 2 we discuss the ontology evolution problem while in Section 3 we discuss a criteria supporting the notion of Ontology Robustness.

2 Ontology Evolution

2.1 The Ontology Evolution Problem

The ontology evolution problem is extremely relevant, as ontology building and annotation are largely manual processes and therefore time-consuming and expensive. For this reason, the risk of invalidating the work done because of the introduction of inconsistencies in evolving the ontology must be reduced as much as possible.

A research tradition that provides notions and tools to deal with change management is in the field of relational and object-oriented database. Here two general approaches were developed. We speak about *schema evolution* when updates are executed without taking care of past versions of the database. Changes are propagated to the data and instances rewriting them according to the specification in the updated schema. We speak about *schema versioning* when changes act on a new version of the schema and a common interface is provided in order to access data that can be stored under different versions of the schema. A general discussion, together with a number of examples about the effects on data models produced by changes in the real world can be found in [17].

Ontology evolution is a comparatively less investigated problem. One of the benchmark papers in this area is [11]. It presents a discussion about the differences between ontology evolution and database schema evolution. The main ones are related to the fact that ontology aims at describing an abstract specification

of a domain, while databases are designed to define the structure of a specific application, and they are seldom intended to be reused. The semantics of a database schema is defined by means of meta-specification, while in the ontology theory the distinction is less definite. Moreover, in many languages for representing ontology the number of representation primitives is much larger than in a typical database schema. Formal semantics of knowledge-representation systems enable us to interpret ontology definitions as a set of logical axioms. For example the OWL-DL language allows the specification of cardinality constraints, inverse properties, transitive properties, disjoint classes, and so on. Relying on logics gives an important motivation to resolve inconsistencies, because an inconsistent logical knowledge-base is by definition ineffective, but at the same time adds an additional layer of specification, increasing the complexity of managing such kind of representation.

The natural evolution of any state of affairs or domain of interest is the crucial fact determining ontology evolution. But even assuming to deal with a static domain an ontology can be evolved to extend its scope to additional domains, or because of a new perspective it is assumed, or simply because of errors discovered in its definition. An important distinction we want to spotlight is among different dimensions of evolution into an ontology. An ontology is both a mental object (an intersubjective conceptualization) and a concrete tool manipulated by humans and machines (a terminology specified in a given language). In more details we distinguish between evolution in:

- **Conceptualization:** the definitions of concepts in an ontology provides its conceptualization. Changes at the conceptualization level act on the concept scope, extension, or granularity.
- **Explication:** depending on the relations between concepts, the same conceptualization can be expressed in different ways. For example, a distinction between two classes can be modeled using a qualifying attribute or by introducing a separate class.
- **Terminology:** one concept can be described by means of synonyms or in terms of different encoding values (for instance distances can be expressed in miles or kilometers).
- **Representation:** different languages, with potentially different expressiveness power, can be used to represent the same conceptualization.

Changes can act on all these dimensions. Moreover, as underlined in [\[8\]](#), changes in one level are not always changes in the other level. In particular we can see a hierarchy of dependencies among dimensions, going from the more general to the more specific. No explication exists without a conceptualization, but a conceptualization can have more than one explication. The problematic implication of this is that detecting a change at a given level does not allow to know if this change is originated by a change at a prior level. More generally, detecting a change among two versions of a same ontology does not allow to understand the motivation triggering this change. This is not just a theoretical problem because, in case of different options to propagate a change, the motivations determine the choice. This apply both to the approaches based on

versioning and change propagation. In the approaches based on versioning this impact on the strategies adopted to query the different versions [6,7]. In the approaches based on change propagation the proposed solutions go through the definition of strategies driving change propagation, according to the directives specified by a designer. In many works these strategies are applied manually by the user [13,9,12]. In [15] a language encapsulating policies for change propagation with respect to ad-hoc strategies was introduced. But this does not prevent a prior work, to be performed by a human agent, to set up strategies. In this paper we propose a solution that allows to manage changes using a single strategy. This will be done at the price of imposing some limitations during the design and annotation phase.

2.2 Migration of Instances

The most critical consequences of ontology evolution is when a change introduces inconsistencies in the knowledge base. This invalidates reasoning and querying tasks and in many architectures can potentially block all the services relying on the ontology. Because the notion of consistency can be easily defined in terms of theoretic semantics, algorithms for consistency checking are widely used. For example in [14] a deep discussion about strategies to maintain an ontology in a consistent state, after propagating changes, is provided. But, as known, an ontology is composed by intensional and extensional knowledge. The intensional knowledge is contained in a set of assertions defining the terminology (Tbox). The extensional knowledge is contained in the assertions using the terminology to annotate specific facts or instances (Abox). Inconsistency applies both to the terminology and the annotations, but in a different way. Inconsistency in the terminology means a contradiction among definitions, while inconsistency in the annotations means a misalignment between the intensional and extensional knowledge. This has a very practical implication in ontology evolution where a relevant problem is to preserve the maximal number of annotations from one version of the ontology to the other. This problem is known as *migration of instances*.

Let us now detail the point with an example. The example is about an informative system which supports an online shop selling books and it is expressed in OWL concrete abstract syntax [18]. In the extract 1 we have the intensional knowledge, defining the terminology used for annotating objects. In the extract 2 we have the annotations. If the assertions in 1 are modified, for instance changing the assertion 11 with a new assertion stating that `bs:Book_of_the_month` is of type `bs:Adventure`, the classification of the annotations in 2 will turn to generate an inconsistency. The problem rises from the lack of a procedure to distinguish appropriate instance assignment for the instance `bs:Book#134`. Is it a `bs:Book_of_the_month` or have it 10% of discount? The restrictive policy will prevent the migration of annotations in 2 to the new version of the ontology. On the contrary, intuition suggests that, if the type assignment is provided by the shop manager this is the annotation to be maintained, while the information about the discount must be updated accordingly. However, any policy for

managing migration of instances is necessary domain-dependent. In the next section we will propose a different solution based on the definition of a criteria allowing to prevent inconsistency and making ontology robust to evolution.

(1)

```

Namespace(bs = <http://ra.crema.unimi.it/ontologies/bookshop#>)
Ontology(
  1. Class(bs:Book partial)
  2. Class(bs:Adventure partial)
  3. Class(bs:Suspence partial)

  4. ObjectProperty(bs:located range(bs:showcase) domain(bs:Book))
  5. ObjectProperty(bs:discounted range(bs:discount) domain(bs:Book))

  6. Individual(bs:5% type(bs:Discount))
  7. Individual(bs:10% type(bs:Discount))
  8. Individual(bs:15% type(bs:Discount))
  9. Individual(bs:20% type(bs:Discount))

  10. Class(bs:Book_of_the_month complete restriction(bs:discount hasValue(bs:15%)))
  11. Class(bs:Book_of_the_month equivalent Class(bs:Suspence))

  ...

```

(2)

```

12. Individual(bs:Book#134 type(bs:Adventure))
13. Individual(bs:Book#134 value(bs:discount bs:10%))

...

```

3 Ontology Robustness

3.1 Ontology in Informative Systems

Let us start recalling the well-known definition by Thomas Gruber: *an ontology is an explicit and formal specifications of a shared conceptualization for some domain of interest* [3]. The notion of a formally specified conceptualization includes a strong relation to the notion of conformity to a norm. The nature of this norm varies and can derive from necessity, regulations, or design goals. The attention that ontologies have achieves in the scientific community is mainly related to this point. Hence, an ontology, when rooted into an informative system, is the tool allowing to enforce a norm into a communicative process. This remark allows to understand that the nature of ontology in today ICT studies is twofold. From one side ontology describes the notions to be enforced. From the other side ontology supports enforcing, which requires the traditional tasks of an informative system: to store and analyze data. In particular the role played by ontologies in data storage is underevaluated. In fact any objet to be inserted

in the memory of a system needs to be recorded according to a set of properties used to identify it. The same happens with ontology; in order to annotate a fact, we need to choose a set of properties identifying it. Understanding this point can help in managing ontologies, in particular in ontology building or in ontology evolution.

3.2 Robustness in Evolution

Intuitively, the notion of *Ontology Robustness in Evolution* is related to the idea of distinguishing among a stable component and a contingent component of the ontology. Knowing such a distinction could allow to design ontologies that can be evolved minimizing the number of changes invalidating instances. The policy to be followed is quite simple. The stable component contains assertions used to annotate facts. The contingent component contains assertions obtained constraining the stable component. Changes can be applied only on the contingent component.

A claim of this paper is that a criterion supporting such a distinction can be derived from the data recording policy used in the informative system. Actually, designing informative systems, it is required to foresee the format in which we will receive data to be stored. This way a schema able to represent these data is recorded in the storage component of the system. For instance, in our example about the book shop one could decide to record books using classical properties such as *author*, *title*, *gender*, et cetera. Such a policy could be adopted in order to distinguish among the stable and contingent components of the ontology. In the example assertions from 1 to 9 represent the stable component of the ontology, while the assertions 10 and 11 represent the contingent component, as they defined a new concept, `bs:Book_of_the_month`, using assertions from the stable component. Then this distinction will drive a policy for managing migration of instances. In particular, referring to the example in Section 2.2, our policy will allow to understand that the annotation 12, specifying the gender of our book, is the one to be maintained, while the annotation about the discount (13) must be updated accordingly.

3.3 Definitions

Let us now summarize our discussion providing some informal definitions specifying the approach we are proposing.

Stable Component. It is a set of assertions of the ontology containing the classes and the properties used to annotate instances. This is a stable component because in general it is not subject to evolution. More precisely changes restricting the knowledge (restricting to the extension of a concept) do not generate inconsistencies [8] and are acceptable even on this component of the ontology.

Contingent Component. It is a set of assertions defining concepts by constraining the assertions in the Stable Component. Using the current standards

for representing ontology, such as for instance OWL-DL, it is easy to write these annotations using restrictions (DL concept constructors).

Ontology Robustness. Informally the notion of Ontology Robustness in Evolution can be explained as the minimization of the number of instances to be migrated in the new version. More formally this notion can be described as the number of changes in the Tbox not invalidation assertions in the Abox. Anyway note that our approach cannot guaranty that inconsistencies are not introduced in the Contingent Component. But in this case we can manage them using traditional algorithms for consistency detection.

Versioning. Our approach is aimed at reducing the use of versioning that is a procedure that can have negative implication in the management of an informative system. Anyway it is clear that versioning cannot be eliminated. For example, due to a radical change in the domain one could deliberately changes the Stable Component, but the designer is now aware that they should potentially invalidate annotations and instances. In this case a typical procedure to follow is to create a new version of the ontology. The assertions not valid in the new version are still valid in the previous one. A positive consequence is that this way versioning plays also the role to mark strong discontinuities in the domain evolution.

4 Conclusions

In this paper we proposed the notion of *Ontology Robustness in Evolution*. This notion applies to ontologies designed according to a criterion that allow to distinguish among a stable component and a contingent component. Where the stable component is used to annotate facts, while the contingent component contains the evolving axioms that classify annotations into the suitable terminology. Further directions of research will point to provide a formal definition of the notion proposed and to investigate its implications in a distributed environment where several dependent ontologies are managed.

Acknowledgements

This work was partly funded by the Italian Ministry of Research under FIRB contract n. RBNE05FKZ2_004, TEKNE.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American 2001(5) (2001), <http://www.sciam.com/2001/0501issue/0501berners-lee.html>
2. Gomez-Perez, A., Manzano-Macho, D.: A survey of ontology learning methods and techniques. OntoWeb Deliverable (2003)

3. Gruber, T.: A translation approach to portable ontology specifications. *Knowledge acquisitions* 5, 199–220 (1993)
4. Hahn, U., Marko, K.G.: Ontology and lexicon evolution by text understanding. In: *Proceedings of the ECAI Workshop on Machine Learning and Natural Language Processing for Ontology Engineering* (2002)
5. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science research in information systems. *MIS Quarterly* 28(1), 75–105 (2004)
6. Heflin, J., Pan, Z.: A model theoretic semantics for ontology versioning. In: *Third International Semantic Web Conference*. Springer, Heidelberg (2004)
7. Huang, Z., Stuckenschmidt, H.: Reasoning with multiversion ontologies: a temporal logic approach. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) *ISWC 2005*. LNCS, vol. 3729, pp. 398–412. Springer, Heidelberg (2005)
8. Klein, M.: *Change Management for Distributed Ontologies*. PhD thesis, Vrije Universiteit Amsterdam (2004)
9. Kalyanpur, A., Parsia, B., Sirin, E., Cuenca-Grau, B., Hendler, J.: SWOOP: A web ontology editing browser. *Journal of Web Semantics* (2005)
10. Malhotra, Y.: *Why Knowledge Management Systems Fail? Enablers and Constraints of Knowledge Management in Human Enterprises*. In: Holsapple, C.W. (ed.) *Handbook on Knowledge Management 1: Knowledge Matters*, pp. 577–599 (2002)
11. Noy, N.F., Klein, M.: Ontology Evolution: Not the Same as Schema Evolution. *Knowl. Inf. Syst.* 6(4), 428–440 (2004)
12. Noy, N.F., Musen, M.A.: The PROMPT suite: Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies* 59(6), 983–1024 (2003)
13. Plessers, P., De Troyer, O.: Ontology change detection using a version log. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) *ISWC 2005*. LNCS, vol. 3729, pp. 578–592. Springer, Heidelberg (2005)
14. Stojanovic, L., Maedche, A., Motik, B., Stojanovic, N.: User-Driven Ontology Evolution Management. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) *EKAW 2002*. LNCS (LNAI), vol. 2473, pp. 285–300. Springer, Heidelberg (2002)
15. Stojanovic, L., Maedche, A., Stojanovic, N., Studer, R.: Ontology evolution as reconfiguration-design problem solving. In: *KCAP 2003*, pp. 162–171. ACM, New York (2003)
16. Vanhoenacker, J., Bryant, A., Dedene, G.: Creating a knowledge management architecture for business process change. In: *The Proceedings of the 1999 ACM SIGCPR conference on Computer personnel research*, New York, NY, USA, pp. 231–241 (1999)
17. Ventrone, V.: Semantic heterogeneity as a result of domain evolution. *SIGMOD Rec.* 20(4), 16–20 (1991)
18. OWL Web Ontology Language Concrete Abstract Syntax. W3C (2003), <http://owl.man.ac.uk/2003/concrete/20031210>
19. OWL - Web Ontology Language definition, <http://www.w3.org/TR/owl-features/>

Learning to Get the Value of Quality from Web Data

Guzmán Llambías, Regina Motz, Federico Toledo, and Simon de Uvarow

Universidad de la República, Facultad de Ingeniería,
Julio Herrera y Reissig 565, 11300 Montevideo, Uruguay
{rmoz, ftoledo, sduvarow, gllambi}@fing.edu.uy

Abstract. The quality of data used in an information system is highly influenced by the quality of data extracted from the sources that the system uses. This feature is particularly sensitive when the sources provide data coming from the web. These web data are extremely dynamic and heterogeneous and they generally lack from a direct responsible about their quality. This work addresses this problem by presenting a proposal to get the values of quality factors from data coming from the web. One important contribution of this paper is the specification of a generic and flexible Quality Factor Ontology (QF-Ontology) able to model quality factors depending not only on the specific application domain but also on the different types of web sources. Moreover, this paper shows how using SWRL the QF-Ontology is exploited to calculate the metrics associated to each quality factor.

Keywords: Data Quality, Ontology, SWRL.

1 Introduction

Currently, the need for techniques to measure and manipulate the quality of data used in information systems is a critical issue. There are several works about data quality, including [1,3,6,7,11]. However, there are no tools yet satisfactory to capture in an automatic way the values of quality. Especially important is this problem in information systems based on the Web, where data quality is highly variable and their quality change in a very dynamic way. Moreover, the definition of “data quality “ depends strongly from the data application domain, the user’s intentions, and the user’s point of view.

This work addresses this problem by presenting a proposal to get the values of quality factors from data coming from the web. One important contribution of this paper is the specification of a generic and flexible Quality Factor Ontology (QF-Ontology) able to model quality factors depending not only on the specific application domain but also on the different types of web sources. We also present the QF-Ontology tailored to the cinemas domain and show how SWRL exploit the use of the ontology.

In order to contextualize the problem, throughout the paper we will be using the following motivating example. We would like to collect and analyzed information about movies. On the one hand, we are interested in movie data: title, actors, directors, and duration. On the other hand, we want information about the movies that are being shown in Montevideo. In order to define our data sources, we chose eleven web

pages about movies and movie theaters in Uruguay. Domain Ontologies are used for enriching the vocabulary with domain knowledge. In this way, we use a wrapper able to identify equivalent terms. The challenge is to get quality values for each web page in order to help us to establish credibility on the different web pages.

The rest of the paper is organized as follows. Section 2 presents the different dimensions (called *factors*) that formed the data quality. Section 3 presents our proposed ontology for the quality factors and their metrics. Section 4 describes how the previous ontology can be used to get the values of quality factors. Finally, Section 5 gives some conclusions and future work.

2 Web Data Quality Factors

The general term "data quality" is a combination of quality factors, where each factor focuses on a specific point of interest. There are several data quality factors: relevance, accuracy, reliability, accessibility, freshness and syntactic correctness are some of them. There are several works that describe each one of these factors [2,6,12]. Which factor is the most relevant, depends on the specific domain of the information system and on the intended use of the data. We use the term "web data quality" in reference to data quality in the special case that the data source is a web page.

In this section we analyze some quality factors specially tailored to the case when data is provided for web pages.

2.1 Web Data Freshness

Data freshness is widely identified as one of the most important factors of data quality to consumers. Some research and empirical studies show that the data freshness is closely linked to the success of an information system. Hence, giving information to the systems about the freshness degree on the web data that they are consuming is a major challenge in the developing of applications. However, there are several definitions about data freshness, in general it is accepted that freshness measures how much updated are data for a specific task. Considering web data, we focus on the concepts of *age* and *volatility* as defined in [2], [6] and [12].

The age suggests how old are data, captures the time interval between the creation or actualization of data and the time at what user receives data. On the other hand, volatility measures the frequency with which data change over time.

Note that it is possible to have updated data and yet they are useless, since their usage time expired. In this sense, the volatility of data is a relevant element in the composition of the freshness degree. For example, if a person needs to know the program of films exhibitions in the cinemas for planning a weekend, the programs may be updated, however, can be made accessible too late and are useless for the period needed.

Therefore, freshness degree is obtained by the metric defined by the formula (1), which relates the concepts of age and volatility.

$\text{Freshness} = \text{Max} \{0, 1 - (\text{age}/\text{volatility})\}$	(1)
---	-----

This metric results in a value between 0 (the data are not fresh) and 1 (data are extremely fresh).

2.2 Web Data Syntactic Correctness

The syntactic correctness concerns that data are free of syntactic errors such as typing errors or format. The data are considered syntactically correct if they comply with user-defined rules or restrictions. Examples of rules are: "classrooms are identified by 3 digits numbers" or the one that says that a date is represented by the format "dd / mm / aaaa", where dd, mm and aaaa are integers such that $00 < dd <= 31$, $00 < mm < 13$, considering also the different number of days according to each month and year (31, 30 or 29). In this example, syntactic correctness verifies that the date is a valid date, without verifying the relationship that the date may have with reality. For example, that a given date is really the date of my birthday. The latter kind of correctness is what is called semantic correctness.

The value of the outcome of the metric that measures the syntactic correctness takes values true or false. A useful idea may be, the use of a range of values more significant than simply Boolean to represent the seriousness of the error. However, the measure of the seriousness of the error varies among different situations depending on the use that will be given to the data, the domain of interest and the viewpoints of the user; making it impossible to establish this range of values generically.

2.3 Web Data Semantic Correctness

The semantic correctness refers to the degree to which data represent real-world. To measure this factor, it is necessary to make a comparison of data with the real world that may be represented by a trusted reference, called oracle, considered always as correct.

Considering, for example a web page that has information on films and their directors, in order to verify whether in the real world each film was directed by the director that is indicated on the page, one should access an external trusted reference in order to evaluate some questions. This reference may be, for instance,, the IMBd-Data [17]. The questions, that this oracle must answer are:

It is correct that <DirectorName> is a director?

It is correct that <FilmName> is a film?

It is correct that <DirectorName> directed the <FilmName>?

This example shows that semantic correctness of data from web pages must be evaluated considering the relationships among all the data. The difficulty lies in that semi-structured data, such as data from a webpage, do not explicitly express these relationships, as they do for instance functional dependencies in relational databases. Therefore, the metric to evaluate this factor depends on the ability to the information extractor to recognize relevant relationships among web data.

Assuming that these relevant relationships among data are known, one possible metrics for this quality factor may be this: If establishing a single question to check its correctness, and this is passed successfully, the page is considered absolutely correct in terms of semantics, evaluated to 1. If establishing only two questions and only one

was successfully approved, the semantic correctness of the web page is evaluated to 0.5 and so on. Ultimately, this is a correct response rate compared to the number of replies.

2.4 Web Data Consistency

The consistency factor measures the fulfillment of integrity restrictions. This factor is strongly related to the semantic correctness factor, if we have data semantically correct, these must be consistent, because they correspond with reality which is always consistent. Anyway, the consistent factor measures complementary aspects from semantic correctness. While the semantics correctness verifies the correctness of the values that take place in relationships within concepts, the consistency factor measures the correctness of relationships itself. For example, whereas the film name and the director name in a web page that really corresponds to the film directed by this director is evaluated as semantically correct, if the data is about a film directed by an animal this is evaluated as an inconsistency.

The detection of this inconsistency is achieved because there exists the restriction that the director of a film must be human. The extractor of data from the web page must be able to validate this restriction from an ontology that describes the domain relations. Hence, the metrics of the consistency factor depends directly from the quality of the data extractor.

3 A Model for Web Data Quality

In order to get the values of quality from web data in a flexible and consistent way, the first step is to specify a formal model that represents the factors involved in the acquisition of the quality of web data as well as the different metrics that can be applied. Our approach to do this challenge is the design of an ontological model inspired in the works of *Qurator* project on biological data quality [4], and also from work in the area of QoS [5]. However, we differentiate from these projects in the sense that our proposal is to model a generic ontology to quality factors (QF-Ontology), independent from the specific domain and from the different types of web data sources. Our generic QF-Ontology in spite of its high level abstraction is easily tailored to different user domains and different types of web data.

In addition to the valuable property of checking the consistency among concepts and relationships, the ontological model provides a high level abstraction that allows specifying in simple way relations between factors and metrics. It also offers a rule language, SWRL [8], which in a declarative way can be used to specify the mechanisms to measure the quality of factors.

Figure 1 depicts, in a simplified way, our proposed ontology for representing the context to assessing quality factors of web data. As described in Section 2 data quality is a composition of quality factors, such as freshness, semantic correctness, consistency, etc. among which holds some relationships. For each quality factor can be defined different metrics that evaluate data and data relationships. Metrics are functions that have input parameters and produce a result. Web data belongs to a web source, which provides metadata. Web source metadata and metrics' parameters

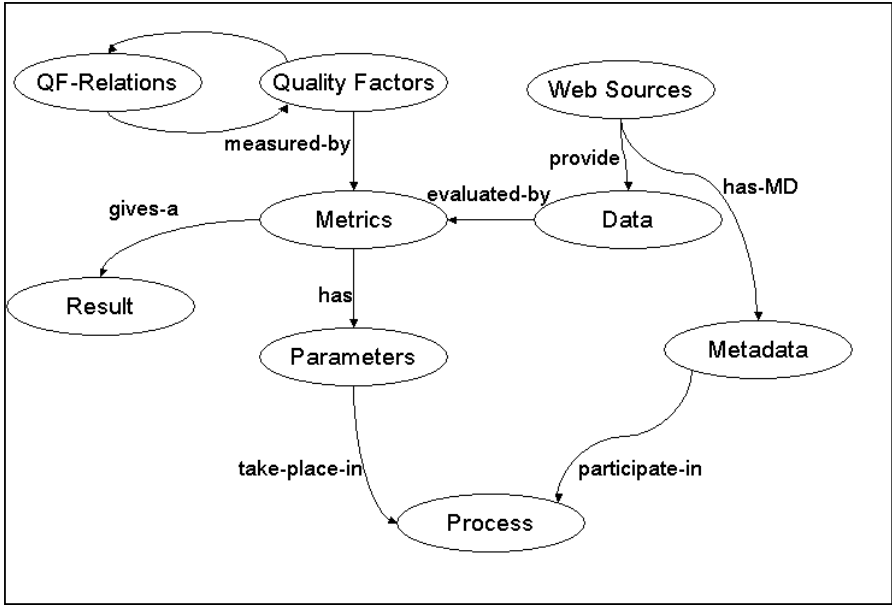


Fig. 1. The Quality Factor Ontology (QF-Ontology)

participate in the process of acquisition of the quality factor values for a given data or relationship between data. The complete Quality Factor Ontology (QF-Ontology) developed in OWL [13] can be accessed through the home page of DiDaWa Project at <http://www.fing.edu.uy/inco/grupos/csi/Proyectos/index.html>

In the QF-Ontology, the Freshness factor for example, is a subclass of QualityFactors and it can be measure using different metrics. If we use the metric proposed in Section 2.1 the input parameters for the movies domain are *age* and *volatility*. The user must give the value of volatility according to the features of the domain, but the value of age must be calculated according to the features of each source. In this sense, web sites have different metadata that can be used to get the value of the age of their data (i.e. headers http, RDF, RSS, ATOM). Therefore, QF-Ontology should be developed for the profile of a user-specific domain and defining the metrics that should be used to get values of each quality factor according to data from specific sources on the web.

In the following section we show this issue and how SWRL is applied to exploit the QF-Ontology to get the values of quality factors.

4 Getting the Values of Quality Factors

In this section we show, by some examples, how the QF-Ontology is tailored to the cinemas domain and how SWRL is applied to get the values of the quality factors.

We begin by defining a subclass of Metrics called *FreshnessMetric*, and then specifying the rule that states the relationship that holds between data and this metric (see Figure 2). A great advantage of using SWRL is its extensibility (by the *Built-Ins*[16]). Basically it consists to implement a function in Java respecting certain restrictions, which will be executed at the time of inference by the rules engine. In this way one can get data to be loaded by the ontology, or compared the parameters with available information in a database, for example.

```

fc:Data(?dat) ∧
swrlx:createOWLThing(?met, ?dat)
→ FreshnessMetric(?met) ∧
fc:data_has_metric (?dat, ?met) ∧
fc:factor_has_metric (Fresh_Inst, ?met)

```

Fig. 2. Rule to assign a metric to data

Following the freshness factor example, it requires the age of the data that can be extracted from the web source by different ways. One way is by the rule depicted in Figure 3, called *getLastModified*. Then, using the metadata values of *CurrentDate* and *LastModified*, the data age is calculated and given to the Freshness Metric as one of the input parameters. The other input parameter, volatility, is defined by the rule depicted in Figure 4, which assign the value of 7 days to the volatility.

```

fc:Source(?f) ∧ url(?f, ?u) ∧
swrlDidawa:getLastModified(?u, ?y) ∧
swrlx:createOWLThing(?m, ?f)
→ fc:has_metadata(?f, ?m) ∧
LastModified(?m) ∧ dateLastModified(?m, ?y)

```

Fig. 3. Rule to get the date of the LastModification

```

fc:Data(?dat) ∧
swrlx:createOWLThing(?vol, ?dat) ∧
Freshness_Metric(?met) ∧
fc:data_has_metric(?dat, ?met)
→ Volatility(?vol) ∧
fc:metric_based_in_attribute(?met, ?vol) ∧
number_of_days(?vol, 7)

```

Fig. 4. Definition of volatility

Once all the input parameters for the metric are defined, we define the process of calculating the metric. This can be seen in Figure 5. The metrics used in the example is defined in this document as $\max(0, (1 - (\text{age} / \text{volatility})))$. This rule creates an instance of *Result* and assigns the value calculated with the input parameters and

```

FreshnessMetric (?met) ∧
Age(?edad) ∧
Volatility(?vol) ∧
fc:metric_based_in_attribute(?met, ?edad) ∧
fc:metric_based_in_attribute(?met, ?vol) ∧
swrlx:createOWLThing(?result, ?met) ∧
number_of_days(?age, ?valAge) ∧
number_of_days(?vol, ?valVol) ∧
swrlb:divide(?resDiv, ?valEdad, ?valVol) ∧
swrlb:subtract(?valorResult, 1, ?resDiv) ∧
swrlDidawa:max(?max, 0, ?valorResult)
→ fc:Result(?result) ∧
fc:metric_has_result(?met, ?result) ∧
fc:quality_value(?result, ?max)

```

Fig. 5. Rule to estimating the FreshnessMetric

inferred as explained above. After having identified all the concepts and rules, the system is set and ready for use.

A complete prototype to get freshness and syntactic correctness quality values from web pages in the cinemas domain can be found at the home page of DiDaWa Project at <http://www.fing.edu.uy/inco/grupos/csi/Proyectos/index.html> The prototype was implemented in Java (JDK 1.5 [14]) and uses the Protege-OWL-API [15] to work with OWL ontologies. In turn Pellet [9] is used to make inferences and system libraries Jess [10] to implement the rules.

5 Conclusions and Future Work

We developed an OWL ontology for Quality Factors that models the problem of getting the values of quality factors from web data in a generic way. Moreover, we show, how this ontology can be tailored, using SWRL, to specific application-domain and user's points of view.

We have shown that quality factors are related, each quality factor has characteristics that differentiate it from others, but usually, the metrics from quality factors are complemented by measuring different aspects of same data. We intend to go ahead in this direction analyzing the impact that some quality factor values may produce in other factors. We believe that this issue may be very useful in order to predict inconsistencies among different definitions of metrics.

We realize that the obtained values of quality factors strongly depend from the wrapper used in the extraction process. This fact is not explicitly considered in the present solution but we are planning to work in this direction.

Aknowledgments

This work was partially supported by grant PDT 41/36.

References

1. Akoka, J., Berti-Equille, L., Boucelma, O., Bouzeghoub, M., Comyn-Wattiau, I., Cosquer, M., Goasdoué-Thion, V., Kedad, Z., Nugier, S., Peralta, V., Sisaid-Cherfi, S.: A Framework for quality evaluation in data integration systems. In: 9th International Conference on Enterprise Information Systems (ICEIS 2007), Funchal, Portugal (Junio 2007)
2. Ballou, D., Wang, R., Pazer, H., Tayi, G.: Modelling Information Manufacturing Systems to Determine Information Product Quality. *Management Science* 44(4) (1998)
3. Lee, Y.W., Strong, D., Kahn, B., Wang, R.: Aimq: A methodology for information quality assessment. *Information and management* 40(2), 133–146 (2002)
4. Manchester; University of Aberdeen. Qurator Project Home Page (Last visited, December 2007), <http://www.qurator.org/>
5. Sena, O., Motz, R.: Hacia un Modelo Genérico para la Calidad de los Servicios Web. II Congreso Español de Setiembre
6. Wang, R., Strong, D.: Beyond accuracy: What data quality means to data consumers? *Journal on Management of Information Systems* 12(4), 5–34 (1996)
7. Wang, R., Storey, Y., Firth, C.: A framework for analysis of data *IEEE Transaction on Data and Knowledge Engineering* 7(4), 623–640 (1995)
8. W3C. SWRL Semantic Web Rule Language (Last visited, November 2007), <http://www.w3.org/Submission/SWRL/>
9. Pellet. OWL DL reasoner (Last visited, November 2007), <http://pellet.owldl.com/>
10. Jess Rule Engine (Last visited, November 2007), <http://herzberg.ca.sandia.gov/>
11. Vaisman, A.: Requirements Elicitation for Decision Support Systems: A Data Quality Approach. In: ICEIS - International Conference on Enterprise Information Systems (2006)
12. Peralta, V.: Data Quality Evaluation in Data Integration Systems. Tesis de Doctorado, Université de Versailles – Universidad de la República, Uruguay (Noviembre 2006)
13. W3C. OWL Web Ontology Language Overview. (Last visited, December 2007), <http://www.w3.org/TR/owl-features/>
14. JDK 1.5. Java Development Kit 5 (Last visited, December 2007), <http://java.sun.com/j2se/1.5.0/>
15. Protégé -OWL API (Last visited, November 2007), <http://protege.stanford.edu/plugins/owl/api/>
16. Built-Ins. para SWRL (Last visited, December 2007), <http://www.daml.org/2004/04/swrl/builtins>
17. Peralta, V.: Extraction and Integration of MovieLens and IMDb Data. Laboratoire PRISM, Technical Report, Université de Versailles, Francia (Junio 2007)

Building the Semantic Utility with Standards and Semantic Web Services

Mathias Uslar¹, Sebastian Rohjans¹, Stefan Schulte², and Ralf Steinmetz²

¹ OFFIS – Institute for Information Technology,
Escherweg 2, 26121 Oldenburg, Germany
{uslar, rohjans}@offis.de

² Multimedia Communications Lab, Technische Universität Darmstadt,
Merckstrasse 25, 64283 Darmstadt, Germany
{Stefan.Schulte, Ralf.Steinmetz}@kom.tu-darmstadt.de

Abstract. Within this contribution, we outline the need for standards in the utility industry. We motivate the need for modern industry standards such as those developed by the International Electrotechnical Commission and introduce concepts based on those standards to foster integration in a modern utility environment by using service-oriented architectures and Semantic Web services. The focus lies on COLIN, a methodology applicable to overcome fallacies in the integration of different industry standards. Furthermore, we propose the usage of information allocated using the COLIN methodology to annotate Web services with semantic information.

Keywords: CIM, Ontology Alignment, Mediators, COLIN, IEC 61970, IEC 61850.

1 Introduction

Several changes in the electric utility domain have imposed new requirements on the IT infrastructures in companies. In the past, the generating structure used to be very closely aligned to the communication infrastructure. Electric energy was delivered top-down from a high-voltage grid having large scale generation attached to a lower-voltage grid and households. The corresponding communication infrastructure was arranged similarly, as control information was mainly passed down the supply chain while data points from the field level were submitted to the SCADA (Supervisory Control and Data Acquisition System).

With the upcoming distributed power generation respectively the legal requirements imposed by federal regulation and the resulting unbundling, things have changed a lot. On the one hand, by deploying new generation facilities like wind power plants or fuel cells, energy is fed into the grid at different voltage levels and by different producers – former customers who have their own power generation now act both as consumers and producers in the utilities’ grid – so called prosumers. Therefore, the communication infrastructure has to change. On the other hand, legal unbundling leads to the separation of systems which have to be open to more market participants.

Hence, this results in more systems which have to be integrated and more data formats for compliance with market participants. The overall need for standards increases. This problem needs to be addressed by an adequate IT-infrastructure within the utility, and supported by architectures like SOA (Service-oriented Architectures). Within this scope, the IEC (International Electrotechnical Commission) has developed data models, interfaces and architectures [4] for running the grid and automating the substations. Unfortunately, these standards have been developed by different working groups and therefore lack fundamental harmonization although they have to be used in context. Furthermore, the semantic techniques imposed by the CIM (Common Information Model) are not properly used. This contribution shows a possible solution for integration based on semantic technologies and provides an outlook for extending the Web services of the IEC TC 57 implementations towards Semantic Web services.

The following contribution is structured as follows. First, we give a brief introduction into the IEC TC 57 standards framework. Afterwards, we show the urgent need to integrate the two main standards within the IEC TC 57 framework in Section 3. For the integration, we introduce the COLIN methodology based on design science methods (Section 4) and provide an overview in one of our use cases developing artifacts for the integration of the two standards IEC 61850 and IEC 61970 (Section 5). Section 6 gives an outline on how to extend the integrated standards in order to foster a Semantic Web services-based integration of all functions for a modern electric utility. Finally, the paper concludes with a summary of the findings in this paper and an outlook of our future work.

2 The IEC TC57 Standards Reference Framework

The IEC has the vision of enabling seamless integration of data for the electric utility domain by using a standards reference framework. Within this standards framework, several standards have been developed by different working groups. Unfortunately, these groups have different ideas regarding what to standardize and the overall focus of their standardization efforts.

Two main standards families exist within the TC 57 framework, the so-called IEC 61970 family, which includes the Common Information Model CIM, and the IEC 61850 family for substation communication which copes with the data exchanged between SCADA systems and field devices. These two families have been developed based on different technical backgrounds.

2.1 The IEC 61970 Family – The Common Information Model CIM

The CIM [3] can be seen as the basic domain ontology for the electric utility domain at the SCADA level. It is standardized in two different sub-families, namely the IEC 61970 family for data model and OPC-based data models which deal directly with the day-to-day business of running an electric power grid. The other is the IEC 61968 family which covers the objects needed to integrate the CIM into an overall utility which has to exchange data with systems like GIS (Geographical Information Systems), CSS (Customer Support System), or ERP (Enterprise Resource Planning).

Overall, the CIM data model covers 53 UML packages which contain roundabout 820 classes with more than 8500 attributes. A lot of effort and work has been put into the model to cover the most important objects for the electric utility domain. Furthermore, different serializations exist. First of all, XML and XML schema exist for building your own Enterprise Application Integration (EAI) messages [2] based on the CIM and use pre-defined messages built by the IEC. Furthermore, RDF serializations and RDF schemas used for modeling the network graphs of power grids for electrical distribution exist, and, based on this work, an overall CIM OWL (DL) serialization has been developed.

Therefore, and due to the maturity of the use of CIM, it can be regarded as one of the largest standardized domain ontologies in effect. Unfortunately, this work was carried out by the IEC working groups WG 13 and 16 which do not standardize the other big family of the IEC TC57 standards reference framework IEC TR 62357, namely the IEC 61850.

2.2 The IEC 61850 Family – Communication for Substations

The IEC TC 57 working group WG 10 has developed the IEC 61850 family which deals with substation automation systems and the corresponding communications. The standard itself is large, and comprises of different kinds of sub-standards like communication protocols, data models, security standards and so on.

The overall domain for IEC 61850 is system automation [1]. While the CIM focuses on energy management systems (EMS), the IT domains are substation intra-application communication and, for the CIM, control-center intra-application communication. Both standards have a basic data model, however the serializations differ severely. While only a small subset for engineering of substations needs a XML-serialization in IEC 61850, all CIM objects can either be serialized using XML, RDF or OWL. Also, the IEC 61850 family lacks significant support from APIs. These differences can result in problems when coping with real-world projects within the IEC TC 57 reference framework. The data model used within the context of IEC 61850 is based on a hierarchical system with a tree-like taxonomy structure and composed data types. Functions have been integrated into the data model with a special focus on set points for control by SCADA systems and data reports have to be implemented using queues and buffers. These two features have not been included in the CIM data model and therefore make harmonization a bit more difficult. The data model for IEC 61850 contains just about 100 classes (so-called logical nodes LNs) which have overall 900 attributes (so-called data objects) from a variety of around 50 base types (so-called common data classes CDCs) consisting of more or less simple types. This figures show that the standard is about as big as the CIM data model mentioned above.

3 The Need for Integration of the IEC Standards

Of course, the data from both standards are used within the same context, therefore, a seamless integration of e.g. the functional description structure of a substation must be known by the SCADA and a mapping between structures from IEC 61850 and IEC

61970 is needed. Furthermore, all data points and measurements from the field devices must be mapped from IEC 61850 semantics to IEC 61970 semantics. Without doubt, these two scenarios are the most striking ones, however further scenarios exist.

Unfortunately, problems have occurred while attempting to use the standards. The different working groups use different naming schemes, object-oriented modeling instead of hierarchical modeling, different semantics, different tools and serializations. Furthermore the IEC 61850 model does not exist as an electronic model, but only within tables in a proprietary text format. Finally, all these standards have been made international standards; therefore they are being implemented by big vendors like ABB, Areva, or Siemens who rely on the stability of the standards for their products. The existing implementations cannot be harmonized at the meta-model level into a new, overall harmonized standard family comprising of IEC 61850 and IEC 61970 without breaking several aspects like proposed by Kostic et al. [1]. Hence, we have to deal with harmonization on the conceptual mediator layer.

In order to cope with all these problems and to facilitate integration, we have developed a methodology for integrating the standards which will be discussed in the next chapter. This method is based on ontology matching algorithms and will develop a bridge-ontology for mediation between the two families of standards.

4 The CIM Ontology aLigNment Methodology (COLIN)

Our “CIM Ontology aLigNment methodology” (COLIN) attempts to overcome all the fallacies described in the previous sections by establishing a methodology for integrating utility standards by taking domain requirements and current research trends into account.

The approach has taken into account the design science approach by Hevner et al. [5] as its scientific method. Our aim is to provide meaningful artifacts to evaluate the use of ontology-based integration and mediation in the context of IEC standards. The artifacts, for example the mediation ontologies created, have been evaluated rigorously and put into different contexts and use cases. We have to distinguish between harmonization on the schema level or at the instance level, for example. The relevance of each artifact being developed must be clear. Furthermore, the ability to transfer the solution found into practice and to show its application to both scientific and managerial authorities is of high importance.

First of all, there are some prerequisites for the approach proposed. We have used the CIM as the basic domain ontology due to its overall size and the objects, attributes and relations already modeled and agreed upon by domain experts. Therefore, the other standards have to be transformed into electronic models in order to be harmonized. We propose the usage of OWL-based ontologies for this task. In order to simplify the mapping process, quantity-based analysis of the standards has been conducted and an overall classification and typology of standards in the utility domain has been created. This data is taken into consideration when attempting to find overlapping parts in the standards which have to be integrated as most cannot be found easily like the ones mentioned above.

Afterwards, the specific parts are modeled using Protégé and serialized in RDF/XML. Now, all the standards have an OWL (DL) representation. We used the

mapping we developed to load the OWL models and to try to find matches based on standard algorithms (structural similarity and phonetic similarity). This procedure led to a number of trivial mappings which had to be verified by domain experts. In order to find more sophisticated mappings, we have developed special domain ontologies and glossaries very similar to approaches such as WordNet – but solely electricity domain based. This process step is the most difficult one and has been supported by domain experts.

Finally, the mapping or mediator ontologies are validated by prototyping and used in production environment, by deploying the ontologies as rule base for EAI based message conversion and pipelining – this leads to the transfer from ontology-based schema integration to instance-based integration.

Currently, we have created an overview of the standards for the utility domain based on several taxonomies. This led to six standards which have to be taken into account when dealing with the IEC TC 57 framework, including the IEC 61850, the IEC 61970 family, the UN/CEFACT CCTS, ebXML, IEC 62361 Quality Codes for Harmonization and the German national grid standard codes. We have developed ontologies for each standard and therefore all electronic models exist. The IEC 61850 family and the CIM are currently being mapped and we have already developed mappings for the UN/CEFACT CCTS and XML naming and IEC 62361 quality codes which have been given to the standards committee to be published as technical reports supporting the standards family.

We briefly introduce the results for the mapping of the IEC 61970 CIM family and the IEC 61850 family based on our ontologies in the next section.

5 Ontology-Based Integration of IEC 61850 and IEC 61970

Using the design science approach described in [5], we have identified several use cases which deal with ontology-based integration [14]. First, we had to identify the standards which had to be integrated, we provided an overview, and developed an ontology to express the concepts and relations the different standards have in general. Second, we had to identify the point at which integration should take place. Here, it was necessary to distinguish between run-time and engineering-time integration. Run-time integration deals with the integration of messages and signal instances and is much more complicated due to complexity and time criticality. The engineering-time integration which basically means integration of schemata or, in our case, integration of ontologies has to deal with larger amounts of data but is not as time critical – it is more tedious but does not need to be performed each time you use the system. The first use case we developed was the integration of data models from the IEC 61970 and IEC 61850 families described in Section 2 of this contribution.

The CIM could already be used as an electronic OWL model as the IEC provides it so. We chose the CIM as our basic domain ontology for the electric energy domain and tried to model all the other standards as OWL ontologies, too. For the IEC 61850 family, the electronic model provided by the national German IEC mirror committee was used and transferred from HTML to the OWL ontology format. Afterwards, we had our two very large domain ontologies which can be obtained for evaluation from [9]. The next step involved choosing the proper way to integrate the two ontologies.

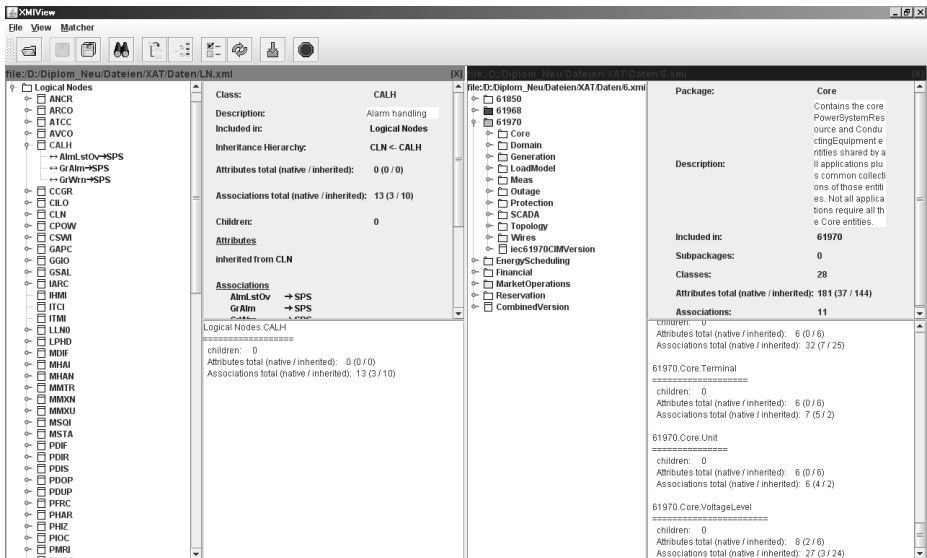


Fig. 1. The Mapping Bench Tool CIMBench

As argued before, international standards cannot be easily changed – implementations exist and have to be taken into account when it comes to drastic changes in the overall models and standards. We chose to integrate the standards with minimal changes to the original data models. One possible solution with minimal impact on the standards is to align them using a mediator ontology [6]. From the beginning, we wanted to use the INRIA alignment format [7] to cope with mappings between the standards we needed. This format is very easy to apply and well supported by current alignment tools like FALCON AO, HMatch or COMA ++.

One of our aims in the integration was to find out whether very large models could be integrated with minimal manual effort, i.e., almost automatically. We tested the three aforementioned mapping tools with different configurations. In most cases the results were not satisfying due to the different structures of ontologies from the standards. Our intention was to keep the OWL ontologies' structure as close to the original standards as possible, thereby reflecting their original hierarchies and depth structure. This has a strong impact on the overall mapping with automated matchers.

Based on the reference mediation ontology created by our domain experts, we got about 30 per cent of all the mappings correct – so we created a specialized mapping tool with a strong focus on the structures of our standards, the so-called CIMBench. The main idea was to focus on string-based, lexical and dictionary-based methods. We integrated all the descriptions from the standards into the data model elements of each individual class in the OWL files and supported them by a dictionary containing the proper descriptions for each term contained in the IEC 60050 electropedia [8]. Furthermore, the original ontologies were analyzed semantically to identify different parts which could be better mapped due to their original purpose have the same focus. The CIM is divided into UML packages whereas the IEC 61850 model is divided into

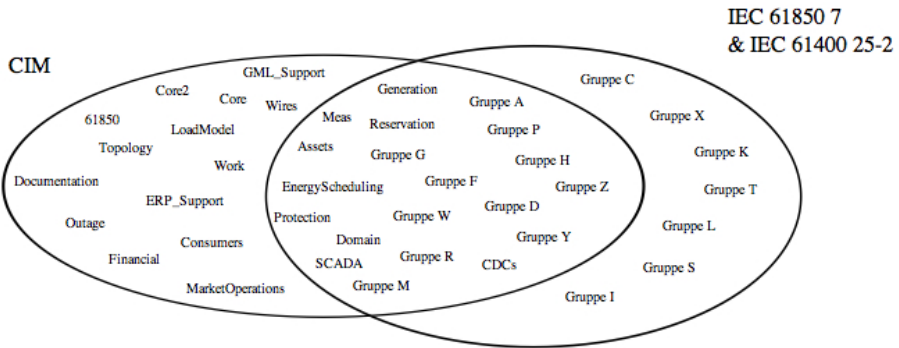


Fig. 2. Intersection between the two standards

twelve groups of nodes which have the same starting letter. This means there are about 26 packages. An analysis performed on the intentions of each package led to the following results for a partitioning shown in Figure 2.

The packages in the middle have been sliced from the large ontologies and have been mapped with our custom CIMBench mapping engine. This led to promising results as a lot of false positives and basic mappings could be ruled out from the resulting alignments. The other packages and elements were mapped afterwards in order to find alignments which were not that obviously based on the basic semantics identified before. Still, packages which were not covered by the intersection contained very important alignments.

As a final result, our custom-made alignment algorithms based on string matching methods like Levenshtein distance, Hamming distance and Jaro-Winkler distance and the input dictionaries combined with a word stemming for the IEC 61850 short names for classes provided promising results. We found about 180 proper alignments out of the 210 estimated by the experts. The latest alignments can be downloaded at [9] or obtained from the authors. In future work, we want to specifically refine the packages and try to find better slicings for the pre-analysis of standards.

By using this harmonized IEC standards as a foundation, we argue that this leads to an integrated data and function model which can be used as a model for semantic annotation for Web services in the electric utility domain. Both standards already have Web services as interfaces, defined like the interface reference model (IRM) in the IEC 61970 CIM family and the Web services interface for 61850 models based on the wind power plant standards and distributed generation standards. However despite this, the strong semantic information imposed by the data models for annotation is not properly used.

6 Instituting Semantic Web Services

As stated in our introduction, the IT infrastructures of companies in the energy domain (as well as in other industries) have to be advanced from communication infrastructures which are purely top-down to much more distributed designs. This can be achieved by

using service-oriented technologies, i.e., Web services that act as distributed entities and provide various types of information from failures of distributed electricity producers to the current wind velocity in a certain area. In general, it is possible to make use of sensors or sensor networks regarding miscellaneous aspects in the energy domain [14]. In addition, databases and other information sources could also be tapped into by designing corresponding Web services. As stated in our previous work [10], a crucial requirement for the automatic retrieval, execution, and composition of Web services is the existence of a clearly defined domain ontology. In the scenario at hand, this ontology is based on industrial standards and provided using the COLIN methodology.

Therefore, it is possible to deploy rich semantic information which can be used in order to annotate Web services with semantic information, thus creating semantic Web services. In this context, the usage of semantic information is an “enabler technology” for innovative application provision and utilization. Instead of providing information respectively Web services through a central entity, e.g., an UDDI-based Web service repository, which might lead to problems in service retrieval, Web services are linked to concepts already known to potential users.

The retrieval of Web services is based on a matchmaking engine, i.e., an algorithm that finds Web services which fit to a query. There is no limit on the technologies used by the matchmaking algorithm, the structure of a request or which service feature should be retrieved. If there were complete and error-free semantic annotations and no lack of information at the parties involved (i.e., service requester or service provider), it would be easy to get matches between a request and existing Web service descriptions. However, this is rather unlikely as it is more likely that at least one of the parties involved does not know how to request or advert a particular service correctly [13]. By “binding” the Web service description in our scenario to the concepts in the domain ontology, possible participators are provided with a data model they use in other settings, too. Thus, it is easier for users to find appropriate Web services.

Apart from using semantic information to align Web services with entities in a power generation domain, and hence, the corresponding service-oriented infrastructure, the semantic information provided could also be deployed to describe the type of messages exchanged, etc.

In order to implement these ideas, we propose the assignment of SAWSDL [11] respectively BPEL for Semantic Web services [12] in order to deploy corresponding Web services. As the name “Semantic annotations for WSDL” implies, SAWSDL adds semantic annotations to WSDL. As WSDL can be taken as the current standard in Web service description languages, SAWSDL is the logical choice for using semantic information created using the COLIN methodology. The work by Nitzsche et al. enables the usage of semantic information in business process descriptions [12]. A tool to semi-automatically annotate Web services with the regarding semantic information is part of our future work.

7 Conclusion and Future Work

Within this contribution, we have provided a short overview of the IEC TC 57 standards framework and its two main standard families. We have discussed the focus of those families and provided a brief account of where fallacies for integration lay,

mainly based on organizational and technical problems. Afterwards, we introduced a description of our COLIN methodology for overcoming these problems by using mediator ontologies and ontologizing standards taking the CIM OWL ontology as our basic domain and upper ontology. In an application case, our solution and the results for the ontology-based integration of the IEC 61970 CIM family and the IEC 61850 standards family have been shown.

The ontology-based integration has really proven useful as mediation is the proper way to integrate existing standards. However, a lack of possibilities for a fully automated mapping outlines the need for either better matching algorithms which take different structures and dictionaries into account, or support by domain experts. By having integrated standards, the harmonized standards offer the possibility to create the SCADA on the Web by using Semantic Web services based on CIM. Thus, it is possible to “bind” Web services to concepts in ontologies. This way, possible parties involved, i.e. service requesters and service providers, are provided with an established and familiar data and infrastructure model.

For our future work, we propose the usage of SAWSDL and BPEL for Semantic Web services. Consequently, it is necessary to provide an application which is able to annotate Web services with semantic information from the ontologies applied. The development of such an application is part of our future work. Furthermore, it would be interesting to transfuse the proposed COLIN methodology to other domains.

References

1. Kostic, T., Frei, C., Preiss, O., Kezunovic, M.: Scenarios for Data Exchange using Standards IEC 61970 and IEC 61850. In: Cigré 2004, Paris. IEEE Publishing, New York (2004)
2. Uslar, M., Streekmann, N., Abels, S.: MDA-basierte Kopplung heterogener Informationssysteme im EVU-Sektor – ein Framework. In: Oberweis, A., Weinhardt, C., Gimpel, H., Koschmider, A., Pankratius, V. (eds.) eOrganisation: Service-, Prozess-, Market-Engineering, 8. Internationale Tagung Wirtschaftsinformatik, Universitätsverlag Karlsruhe, vol. 2 (2007)
3. IEC – International Electrotechnical Commission: IEC 61970-301: Energy management system application program interface (EMS-API) – Part 301: Common Information Model (CIM) Base. International Electrotechnical Commission (2003)
4. Robinson, G.: Key Standards for Utility Enterprise Application Integration (EAI). In: Proceedings of the Distributech 2002, Miami, Pennwell (2002)
5. Hevner, A.R., March, S., Park, J., Ram, S.: Design Science Research in Information Systems. Management Information Systems Quarterly 28, 75–105 (2004)
6. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Heidelberg (2007)
7. Shvaiko, P.: An API for ontology alignment, Version 2.1 (2006), <http://gforge.inria.fr/docman/view.php/117/251/align.pdf>
8. IEC – International Electrotechnical Commission: IEC Electropedia, <http://www.electropedia.org>
9. OFFIS: Ontologies for the utility domain (2008), <http://www.offis.de/energie/ontologies>
10. Schulte, S., Eckert, J., Repp, N., Steinmetz, R.: An Approach to Evaluate and Enhance the Retrieval of Semantic Web Services. In: 5th International Conference on Service Systems and Service Management, ICSSSM 2008 (2008)

11. Farrell, J., Lausen, H.: Semantic Annotations for WSDL and XML Schema. W3C Recommendation (2007), <http://www.w3.org/TR/2007/REC-sawsdl-20070828/>
12. Nitzsche, J., van Leesen, T., Karastoyanova, D., Leymann, F.: BPEL for Semantic Web services. In: Proceedings of the 3rd International Workshop on Agents and Web services in Distributed Environments, AWeSome 2007 (2007)
13. Paolucci, M., Kawamura, T., Payne, T., Sycara, K.: Semantic Matching of Web services Capabilities. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 333–347. Springer, Heidelberg (2002)
14. Uslar, M.: Ontology-based Integration of IEC TC 57 standards. In: Workshop proceedings of the I-ESA 2008 conference, Fraunhofer IPK Berlin, pp. 31–34 (2008)

TICSA Approach: Five Important Aspects of Multi-agent Systems

Maja Hadzic and Elizabeth Chang

Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology
GPO Box U1987, Perth 6845, Australia
{m.hadzic,a.sidhu,t.dillon}@curtin.edu.au

Abstract. The issues of distributed and heterogeneous information, lack an underlying knowledge base, autonomy of the information resources, dynamic nature of the information retrieval process and dramatically increase of the available information are major factors hindering efficient and effective use of the available information. In this paper we explain the importance of multi-agent systems in addressing these issues effectively and illustrate this on an example of multi-agent system designed to intelligently retrieve human disease information. We also present a conceptual framework (TICSA) which focuses on the five different aspects of multi-agent systems namely, agent Types, Intelligence, Collaboration, Security and Assembly. This framework can be used to provide insight and guidance during the multi-agents systems design.

Keywords: multi-agent system, multi-agent system design, human diseases, information retrieval, biomedical information systems.

1 Introduction

Agents are intelligent programmes used for perform various actions. They can answer queries, retrieve information, make decisions and communicate with computer systems, other agents or users. Agents are capable to perform their actions autonomously and are sociable, reactive and proactive in an information environment [1]. The main features of agents are their autonomous, intelligent, mobile, cooperative and collaborative capabilities. The main operations of a multi-agent system are based on effort of collaborative working agents; different types of agents are working cooperatively towards a shared goal. The multi-agents systems greatly contribute to the design and implementation of complex biomedical information systems.

Effective implementation of multi-agent systems within biomedical domain could result in a revolutionary change that will positively transform the existing biomedical system. The main issues hindering effective use of the available information include [2]:

1. size of the available information
2. autonomous, distributed and heterogeneous nature of the information resources, and
3. lack of tools to analyse the available information and derive useful knowledge from it.

The users are faced with additional difficulties which include [2,3]:

- a) rapid increase of medical information (new papers or journals are being published with a high rate)
- b) inconsistent structures of the available information (as a result of autonomy of information resources)
- c) related, overlapping and semi-complementary information
- d) existence of complex diseases e.g. mental illnesses or diabetes. The complex diseases are caused by a number of genes usually interacting with various environmental factors [4].

In this paper we propose multi-agent systems as a solution to those problems. Related work is discussed in Section 2. In Section 3, we discuss the design of multi-agent systems used to intelligently retrieve information about human diseases. Each subsections of the Section 3 correspond with a specific aspect of the TICSA generic conceptual framework that can be used to guide the system design. We give our final remarks in Section 4.

2 State of Play

Multi-agent systems are being used more and more in the medical domain. Some of these agent-based systems are designed to use information within specific medical and health organizations, others use information from Internet.

The information available to organization-based systems is limited to a specific institution and these multi-agent systems help the management of the already available information. They do not have a purpose of gaining new knowledge regarding the disease in question. For example, Agent Cities [5] is a multi-agent system composed of agents that provide medical services. The multi-agent system contains agents that allow the user to search for medical centres satisfying a given set of requirements, to access his/her medical record or to make a booking to be visited by a particular kind of doctor. AACare [6] agent architecture is a decision support system for physicians. It connects patient's records with the predefined domain knowledge such as knowledge regarding a specific disease, a knowledge base of clinical management plans, a database of patient records etc. MAMIS [7] is a Multi-Agent Medical Information System facilitates patient information search and provides ubiquitous information access to physicians and health professionals.

Other multi-agent systems retrieve information from the Internet. BioAgent [8] is a mobile agent system where an agent is associated to the given task and it travels among multiple locations and at each location performs its mission. At the end of the trip, an information integration procedure takes place before the answer is deployed to the user. Holonic Medical Diagnostic System [9] architecture is a medical diagnostic system that combines the advantages of the holonic paradigm, multi-agent system technology and swarm intelligence in order to realize Internet-based diagnostic system for diseases. All necessary/available medical information about a patient is kept in exactly one comprehensive computer readable patient record called computer readable patient pattern (CRPP) and is processed by the agents of the holarchy. Different web crawling agents [10] have been designed to fetch information about diseases when given information about genes that when mutated may cause these diseases.

The importance of use of the multi-agent system within a specific institution such as hospital or a medical centre is great. In this project we focus on a different level of contribution, namely, on making a channel through which newly available and valid information from the research arena will flow into the medical practice to be effectively implemented there. Lots of the information is available but due to the large body of information some important information may escape the users notice and be neglected.

The BioAgent system could be used by our system with some modifications. We can use the same principle of agent migration among multiple locations, information retrieval from each location and information integration at the end of the trip. Only the information we are interested in is not regarding the genome analysis and annotation but human diseases. There is need to design a multi-agent system for the purpose of dynamic information retrieval regarding common knowledge of human diseases as such a system does not exist yet. Holonic Medical Diagnostic System is Internet-based system but it operates on the basis of the information specified in the patient record and collecting the evidence for diagnosis of this patient. Web crawling agents focus only on genetic causes of human diseases. In this project, we propose a system that integrates information regarding disease types, symptoms, causes and treatments of a disease in question. This multi-faceted approach is very significant in the domain of complex diseases where a specific combination of genetic and environmental factors causes a specific type of a specific disease.

3 TICSA Approach

In this section, we describe how we use the Agent Design Methodology described in [11] to design a multi-agent system for retrieval of information about human diseases. The Agent Methodology consists of the following five steps:

1. Identify Agent Types According to Their Responsibilities
2. Define Agent's Intelligence
3. Define Agent's Collaborations
4. Protect the System by Implementing Security Requirements
5. Assemble Individual Agents

The key aspect of each step is represented in Figure 1.

3.1 Identify Agent Types According to Their Responsibilities

A multi-agent system is a community of agents. The agents are characterized by different but complementary capabilities and are cooperatively working towards the shared goal. The agents are required to work in unity, coordinate their actions and integrate their results.

When identifying agent types, it is important to:

- establish intuitive flow of problem solving, task and result sharing
- identify agent functions needed to establish this kind of flow
- identify agent roles according to these functions

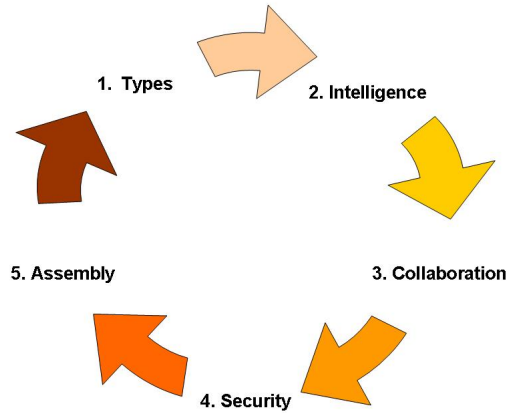


Fig. 1. Diagram representing main focus of each Agent Design Methodology step

In our example, a user queries the multi-agent system and the multi-agent system answers the query in an intelligent way. A range of actions is required to provide the user with correct answer. These include (1) translation of user's query into a machine-understandable language, (2) sharing of the information retrieval task between different agents, (3) activation of appropriate agents to retrieve the target information, (4) analysis of the retrieved information, (5) selection of appropriate information, (6) assembly of the selected information, and (7) presentation of the assembled information to the user.

We have identified four agent types required to fulfil the overall task of intelligent information retrieval. The organization of the different agent types within the information system is presented in Figure 2. All agents within this information system are dependent on each other for the realization of the shared goal. Their common goal is to answer the user's query in the most efficient way. To be able to achieve this, they have different functions and work on different levels within the multi-agent system. In this distributed multi-agent system architecture, *Interface agents* assist users in formulating queries as well as in presenting assembled information back to the user. Interface agents communicate user's request to *Manager agents*. *Manager agents* then assign specific tasks to *Information agents*. *Information agents* retrieve requested information from a wide range of biomedical databases. Each Information agent may have a set of assigned databases that it needs to visit in order to retrieve requested information. Information agents hand over retrieved information to *Smart agents*. *Smart agents* analyze this information, select relevant information, assemble it correctly and pass this information to *Interface agents* to be presented to the user as an answer to his/her query.

3.2 Define Agent's Intelligence

The agents need to be equipped with the knowledge that will enable them to perform their task intelligently e.g. to decompose overall problem, to retrieve relevant information, to communicate with each other, to analyze and manipulate information, present information in a meaningful way, etc. Currently, knowledge bases have been

predominantly used to provide agents with intelligence and enable them to perform their action efficiently and effectively. Ontologies are high expressive knowledge models and use of ontologies over knowledge bases is preferred [12].

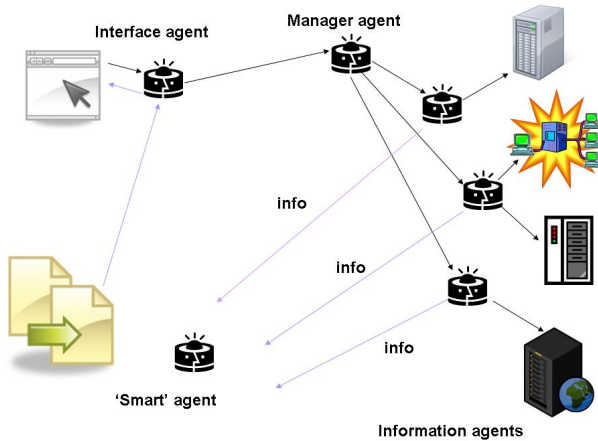


Fig. 2. Interface, 'Manager', Information and 'Smart' agents

In our previous works [13], we have explained design of Generic Human Disease Ontology (GHDO) as being composed of four sub-ontologies: Disease Types, Symptoms (Phenotype), Causes and Treatments (see Figure 3). This ontology can be used to equip agents with intelligence and enable them to retrieve relevant information intelligently.

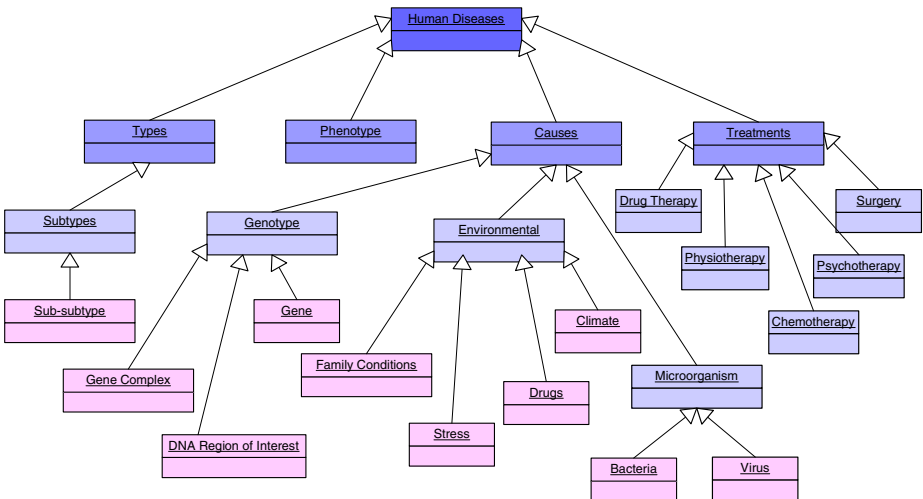


Fig. 3. Four subontologies of the Generic Human Disease Ontology

In the sequel of this section, we will explain how the GHDO can be used increase intelligence as well as control over the information retrieval process. Interface agent maps the user's query to GHDO concepts, and assembles the mapped GDHO concepts into a Specific Human Disease Ontology template (SHDO template). This SHDO template is subset of the GHDO, and is a template into which the retrieved information will be filled in. To enable effective problem decomposition and task sharing among different agents, Manager agent decompose the SHDO template according to the four sub-ontologies and assigns relevant tasks to appropriate Information Agents. Information agents retrieve the target information and pass it over to Smart agent. Smart agent analyzes this information, selects relevant information and assembles it into the SHDO template. This step results in a Specific Human Disease Ontology (SHDO) that is presented to the user as the answer to his/her query. With the focus on the ontology, this process is shown in Figure 4.

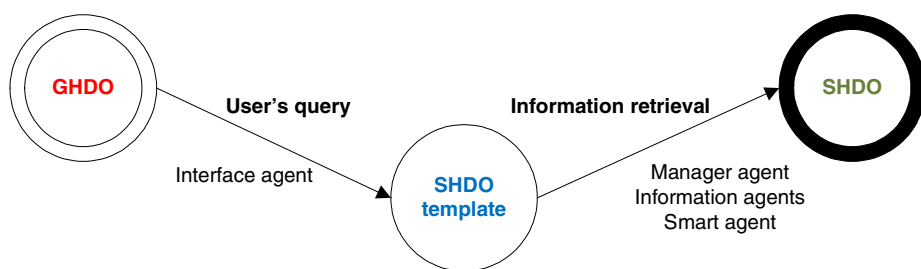


Fig. 4. GHDO, SHDO template and SHDO

3.3 Define Agent's Collaborations

In the first stage of the TICSA approach, we described how to identify different agent types according to their different functions and roles within the multi-agent system. In this stage, we focus on structural organization and position of agents within the system. The aim of this step is to:

- define system architecture that will enable the most optimal performance of agents
- establish correspondence between different agent types and positions of these agents within the multi-agent system

Here it is important to organize the agents so that the problem solving process can easily flow towards its completion and that the communication between different agents can be easily established. In combination with capabilities of individual agents, these two features are major factors determining efficient and effective system performance. Sometimes, a system structured in a simple way functions the best. In other cases, a complex system may be a better choice for the task at hand.

We have proposed a GHDO-based Holonic Multi-agent Structure (GHMS) [14] (shown in Figure 5) as a nested hierarchy of four holarchies in which each of the four GHDO dimensions (Disease types, Symptoms, Causes and Treatments) is associated with one holarchy. The information is interpreted and analyzed at the higher levels of the hierarchy while collection of the data happens at the lower level of the holarchy.

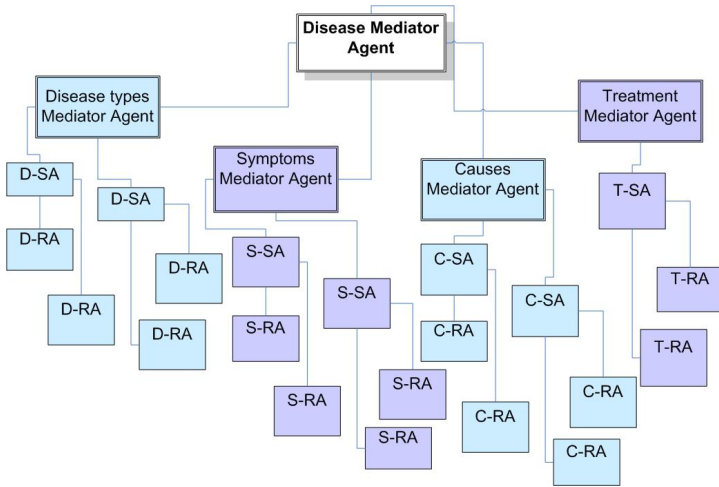


Fig. 5. GHMS structure

Highest in the agent hierarchy is Disease Mediator Agent. For each of the four holarchies, we have corresponding Mediator Agent, Specialist and Representative Agents.

Disease Mediator Agent (DMA) is the main entry point of GHMS. It also functions as Interface agent and creates SHDO template according to user’s query. On the basis of the SHDO template, DMA decides which of the four holarchies needs to be activated. For example, sometimes a user may be interested only in the treatments of a disease so that there is no need to deploy the Disease types, Symptoms and Causes holarchy. DMA also corresponds to the first level Manager agent.

Mediator Agents (MAs). Each branch of the main entry point of GHMS (DMA) has its own mediator agents, respectively Disease Types, Symptoms, Causes and Treatments Mediator Agents (D-MA, S-MA, C-MA and T-MA). Their task is to decide which other Specialist Agents (SAs) need to be activated to retrieve requested information. MAs correspond to the second level Manager agents.

Specialists Agents (SAs). Holarchy inner nodes represent Specialist Agents (SAs). We differentiate Disease types, Symptoms, Causes and Treatments Specialists Agents (D-SA, S-SA, C-SA and T-SA). They are specialists on a specific topic of corresponding subontology. For example, within Causes sub-ontology one C-SA may be specialized in the genetic causes of a disease while another C-SA may be a specialist on the environmental causes of a disease.

SAs assign different tasks to different RAs. SAs correspond to the third level Manager agents. After RAs have returned their data, SAs interpret, analyse, select and assemble these data into relevant part of the SHDO template. SAs correspond to the third level Smart agents.

SAs pass the assembled information onto MAs which also receive assembled information from other SAs of the same sub-ontology. The MAs correspond to the second level Smart agents and they analyze all this information, select relevant

information and merge it into the SHDO template. At the end of this process, the SHDO template contains information relevant to specific sub-ontology. This information is forwarded to DMA.

DMA receives the four SHDO sub-ontologies which contain information regarding disease types, symptoms, causes and treatments, merges this complementary information into SHDO and present this information to the user. MAs correspond to the first level Smart agent.

Representative Agents (RAs). The leaves are so-called Representative Agents (RAs). We differentiate Disease types, Symptoms, Causes and Treatments Diseases Representative Agents (D-RA, S-RA, C-RA and T-RA). Each RA is an expert on the lowest level concept within the ontology. Note that RAs differ from SAs in that they need to recognize the significant information inside the appropriate database and retrieve that information. RAs correspond to the Information agents.

3.4 Protect the System by Implementing Security Requirements

Security plays an important role in the development of multi-agent systems. The risks of jeopardizing the system security must be minimized by providing as much security as possible. The aim of this stage is to:

- identify critical security issues within the multi-agent system
- effectively address the identified issues
- implement the security requirements within the system

In the GHMS environment, all five security properties [15] of authentication (proving the identity of an agent), availability (guaranteeing the accessibility and usability of information and resources to authorized agents), confidentiality (information is accessible only to authorized agents), non repudiation (confirming the involvement of an agent in certain action) and integrity (information remains unmodified from source entity to destination entity) should be taken into consideration. Additional agent's characteristics such as compliance (acting in accordance with the given set of regulations and standards), service (serving one another for mutually beneficial purposes) and dedication (complete commitment of the agents to the overall goal and purpose of the multi-agent system), greatly contribute to the security of the overall system.

The abovementioned properties are critical inside the multi-agent system as well as outside the multi-agent system, such as during the agent interaction with the environment. After the identification of required security properties, it is necessary to effectively address and implement them within the multi-agent system. As different agents have different functions within the system, some agents will be more critical than others in regard to the security of the system. As a consequence, the critical agents will be assigned more security requirements than the others.

3.5 Assemble Individual Agents

In the previous sections, we have discussed functions of agents within a system as well as equipping the agents with intelligence and enabling them to perform these functions optimally, collaborative aspect of agents and security requirements. In this

step we focus on bringing these different aspects together and creating a variety of agents. For each agent, it is important to:

- identify required agent components
- assemble the components into an unified system i.e. individual agents

These agent components may include the Human interface (interaction with users), Agent interface (interaction with agents), Communication component (processing messages), Cooperative (negotiation, cooperation and coordination), Procedural (problem solving procedures, goal prioritization), Task (agent-specific), Domain (domain of interest) and Environment knowledge (in which the agent is situated), History files (past experiences), and so on.

We have chosen the assembly to be the last design step as many different agents will have a number of components in common. The variety of agents within a multi-agent system can be achieved in three different ways: (1) different components that are used to construct different agents are the same, but the *content* of these components is different for different agents, (2) the content of the components used to construct different agents are the same, but different agents are constructed by a different *combination* of used components and (3) the third and most common option is that different agents differ in the *combination* of the components used to construct them, *and* in the *content* of these components.

4 Conclusion

In this paper, we discussed current issues associated with the access, storage, management and retrieval of biomedical information and proposed multi-agent system as possible solution to these problems. We discussed design of the multi-agent systems using the TICSAs (Types, Intelligence, Collaboration, Security and Assembly) approach and illustrated the idea on the design of ontology-based multi-agent system for retrieval of information about human diseases.

The implementation of multi-agent systems within health and medical domain on a larger scale will result in positive transformation of world-wide health and medical research and management to a more effective and efficient regime.

References

1. Wooldridge, M.: An Introduction to Multiagent Systems. John Wiley and Sons, Chichester (2002)
2. Hadzic, M., Chang, E.: Web Semantics for Intelligent and Dynamic Information Retrieval Illustrated Within the Mental Health Domain. In: Advances in Web Semantics. Springer, Heidelberg (2007)
3. Goble, C.: The Grid Needs you. Enlist Now. In: Proceedings of the International Conference on Cooperative Object Oriented Information Systems, pp. 589–600 (2003)
4. Smith, D.G., Ebrahim, S., Lewis, S., Hansell, A.L., Palmer, L.J., Burton, P.R.: Genetic epidemiology and public health: hope, hype, and future prospects. *The Lancet* 366(9495), 1484–1498 (2005)

5. Moreno, A., Isern, D.: A first step towards providing health-care agent-based services to mobile users. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multi-agent Systems, pp. 589–590 (2002)
6. Huang, J., Jennings, N.R., Fox, J.: An Agent-based Approach to Health Care Management. *International Journal of Applied Artificial Intelligence* 9(4), 401–420 (1995)
7. Fonseca, J.M., Mora, A.D., Marques, A.C.: MAMIS – A Multi-Agent Medical Information System. In: Proceedings of IASTED International Conference on Biomedical Engineering, BioMED 2005 (2005)
8. Merelli, E., Culmone, R., Mariani, L.: BioAgent-A mobile agent system for bioscientists. In: Proceedings of the Network Tools and Applications in Biology Workshop Agents in Bioinformatics (2002)
9. Ulieru, M.: Internet-enabled soft computing holarchies for e-Health applications. *New Directions in Enhancing the Power of the Internet*, 131–166 (2003)
10. Srinivasan, P., Mitchell, J., Bodenreider, O., Pant, G., Menczer, F.: Web Crawling Agents for Retrieving Biomedical Information. In: Proceedings of International Workshop on Agents in Bioinformatics, NETTAB 2002 (2002)
11. Hadzic, M., Chang, E.: Onto-Agent Methodology for Design of Ontology-based Multi-agent Systems. *International Journal of Computer Systems Science and Engineering* 22(6), 65–78 (2007)
12. Maedche, A.D.: *Ontology Learning for the Semantic Web*. Kluwer Academic Publishers, Norwell (2003)
13. Hadzic, M., Chang, E.: Ontology-based Multi-agent systems support human disease study and control. In: Czap, H., Unland, R., Branki, C., Tianfield, H. (eds.) *Frontiers in Artificial Intelligence and Applications (special issues on Self-organization and Autonomic Informatics)*, vol. 135, pp. 129–141 (2005)
14. Hadzic, M., Ulieru, M., Chang, E.: Soft Computing agents for e-Health Applied to the Research and Control of Unknown Diseases. *Information Sciences (special journal issue on Softcomputing meets Agents)* 176, 1190–1214 (2006)
15. Mouratidis, H., Giorgini, P., Manson, G.A.: Modelling secure multi-agent systems. In: Second International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 859–866 (2003)

Applications of the ACGT Master Ontology on Cancer

Mathias Brochhausen¹, Gabriele Weiler², Luis Martín³, Cristian Cocos¹,
Holger Stenzhorn⁴, Norbert Graf⁵, Martin Dörr⁶, Manolis Tsiknakis⁶,
and Barry Smith^{1,7}

¹ IFOMIS, Saarland University P.O.15 11 50, 66041 Saarbrücken, Germany

² Fraunhofer Institute for Biomedical Engineering, St. Ingbert, Germany

³ Biomedical Informatics Group, Artificial Intelligence Laboratory, School of Computer Science, Universidad Politécnica de Madrid, Madrid, Spain

⁴ Institute of Medical Biometry and Medical Informatics, University Medical Center, Freiburg, Germany

⁵ Paediatric Haematology and Oncology, Saarland University Hospital, Homburg, Germany

⁶ Foundation for Research and Technology-Hellas (FORTH), Institute of Computer Science, Heraklion, Greece

⁷ Department of Philosophy and New York State Center of Excellence in Bioinformatics and Life Sciences, University at Buffalo, USA

mathias.brochhausen@ifomis.uni-saarland.de

Abstract. In this paper we present applications of the ACGT Master Ontology (MO) which is a new terminology resource for a transnational network providing data exchange in oncology, emphasizing the integration of both clinical and molecular data. The development of a new ontology was necessary due to problems with existing biomedical ontologies in oncology. The ACGT MO is a test case for the application of best practices in ontology development. This paper provides an overview of the application of the ontology within the ACGT project thus far.

Keywords: biomedical ontology, clinical trials, mediation.

1 Introduction

Over the last decade the amount of data on cancers and their treatment has exploded due to advances in research methods and technologies. Recent research results have changed our understanding of fundamental aspects of cancer development at the molecular level. Nevertheless, irrespective of the fact that huge amounts of multilevel datasets (from the molecular to the organ and individual levels) are becoming available to biomedical researchers, the lack of a common infrastructure has prevented clinical research institutions from being able to mine and analyze disparate data sources efficiently and effectively. As a result, very few cross-site studies and multicentric clinical trials are performed, and in most cases it is not possible to seamlessly integrate multi-level data. Moreover, clinical researchers and molecular biologists often find it hard to take advantage of each other's expertise due to the absence of a

cooperative environment which enables the sharing of data, resources, or tools for comparing results and experiments, and of a uniform platform supporting the seamless integration and analysis of disease-related data at all levels [1]. This situation severely jeopardizes research progress and hinders the translation of research results into benefits to patients.

The Advancing Clinico-Genomic Trials on Cancer (ACGT) integrated project aims to address this obstacle by setting up a semantic grid infrastructure in support of multi-centric, post-genomic clinical trials [2]. This system is designed to enable the smooth and prompt transfer of laboratory findings to the clinical management and treatment of patients. Obviously, this goal can only be achieved if state-of-the-art semantic technologies are part of the IT environment. In order to meet this goal, the ACGT project needed an ontology to be utilized in the context of its selected Local-As-View (LAV) data integration strategy [3]. In such a strategy the ontology plays the role of a global schema to which all local schemata are mapped, so that all their mapped equivalents are subsumed by the global schema. This requires that the global schema (i.e. the ontology) be sufficiently generic as to cover not only terminology, but also the meaning of all local schema constructs. The ACGT project achieves the semantic integration of heterogeneous biomedical databases through a service oriented, ontology driven mediator architecture that makes use of the ACGT-MO [4, 5]. The new terminology resource which underlies this integration rests upon a thorough review and critical assessment of the state of the art in semantic representation of cancer research and management.

2 Pre-existing Ontologies and Terminologies

Cancer has been a focus of interest in biomedical research for a very long time. As a result of this long history, a number of terminological resources exist that are of relevance to ACGT. In order to prevent redundancy, the project undertook a very detailed review. We will illustrate this selection process by focusing on two potential resources that did not meet our criteria of excellence, and hence were either not used in ACGT, or were used after considerable alteration. We will further mention two general biomedical resources selected for integration in the ACGT terminological network.

When considering the development of an ontology-based information-sharing system for the cancer domain of the sort used by ACGT, the National Cancer Institute Thesaurus (NCIT) is a terminology resource of obvious relevance [6]. Yet, there are a number of drawbacks preventing the use of the NCIT as semantic resource of the ACGT project, in part because its formal resources are too meager for our purposes, with only a fraction of NCIT terms being supplied with formal definitions of the sort required by its official description logic (DL) framework. The NCIT contains only one relation, namely the subtype relation (*is_a*), as contrasted with the plurality of formally defined relations included, for example, within the OBO Relation Ontology [7]. Further, the NCIT is marred by a number of problems in its internal structure and coverage [8], including problems in the treatment of *is_a*. For a quick illustration of the inadequate treatment of *is_a* in the reviewed version of NCIT, let us consider the NCIT class *Organism*, which includes among its subtypes *OtherOrganismGroupings*; with this we have *OtherOrganismGroupings is_a Organism* [6]. Given the formal

definition of the subtype relation this is clearly wrong; groupings of organisms are not themselves organisms.

Another resource that has the aura of indispensability in a domain dealing with gene array data is the Microarray and Gene Expression Data (MGED) ontology [9]. Yet, even this highly used resource shows considerable deficiencies, including informal *is_a* relations. The inconsistency becomes obvious when the textual definitions – which are an asset to MGED – are taken into account: According to the MGED ontology *Host* is a subclass of *EnvironmentalHistory*. It is obvious that this cannot be a formal *is_a* relation. Taking a close look reveals an astonishing incoherence here: The definition of *Host* is: “Organisms or organism parts used as a designed part of the culture (e.g., red blood cells, stromal cells)” [9]. The definition of *EnvironmentalHistory* reads as follows: “A description of the conditions the organism has been exposed to that are not one of the variables under study” [9]. The thesis that an organism or organism part is a description clearly involves a crude category mistake (the confusion of use and mention). For some portions of the ACGT domain, however well-built and well maintained ontologies with high usability could be identified and reused within ACGT. This, as a matter of fact, applies both to the Foundational Model of Anatomy (FMA) [10] and the Gene Ontology (GO) [11] since they both fulfill the requirements on coherence and theoretical rigor specified in [12].

Most of the current ontologies for life sciences start from terminology appearing in documentation systems as data and pertaining to the “subject matter” of the research carried out, such as concepts about the human body, diseases and microbiological processes. However, the data kept in the systems ACGT aims at supporting also pertain to the scientific processes of observation, measurement and experimentation together with all contextual factors. A model for integrating that data must include this aspect. The CIDOC CRM (ISO21127, [13]) is a core ontology originally developed for schema integration in the field of documenting the historical context and treatment of museum objects, including a generic model of scientific processes. Some concepts and relations of the latter were reused and refined for the ACGT MO.

Effectively, developing a new ontology was imperative, since no single ontology or set of ontologies had the respective coverage and logical consistency.

3 The ACGT Master Ontology

3.1 Technical Details of the ACGT MO

The intention of the ACGT MO [4] is to represent the domain of cancer research and management in a computationally tractable manner. As such, we regard it as a domain ontology. The initial version of the ACGT MO that was made public on the internet consists of 1300 classes. The ontology was built, and is being maintained, using the Protégé-OWL open-source ontology editor [14]. It is written in OWL-DL [15] and presented as an .owl file. The ACGT MO not only represents classes as linked via the basic taxonomical relation (*is_a*), but connects them via other semantic relations called “properties” in OWL terminology. The OBO Relation Ontology (RO) [7] has been used as a basis in this regard, as RO has been specifically developed to account

for relations in biomedical ontologies [16]. Some properties of scientific observation were taken from the CIDOC CRM [13].

3.2 Methodology

The ACGT MO has been developed in close collaboration with clinicians utilizing existing Clinical Report Forms (CRFs), which were used to gather documentation on the universals and classes in their respective target domains, and to understand the general semantics of form-based reporting of clinical observation. All versions of the ontology have been reviewed by clinical partners who have proposed changes and extensions according to needs. In this process the problem of handling an ontology with more than 1300 classes for clinical users became apparent. Providing tools to examine the ontology in user-friendly ways emerged as inevitable. Yet, to ensure comprehensiveness of the representation of relevant portions of reality it was found necessary to go beyond the CRFs and the documentation provided by the clinical project partners. The latter governed the development of the leaf nodes of the ACGT MO, but we had to identify classes for a middle layer of the representation in order to ensure that the ontology provided the necessary reasoning support. Therefore, standard literature and standard classification systems were used, e.g. [17, 18, 19]. In order to provide a consistent and sound representation, the ACGT MO employs the resources of an Upper Ontology, which does not represent domain specific knowledge, but consists of classes that are generic and abstract [20]. The ACGT MO is based on Basic Formal Ontology (BFO) [21] which has proven to be highly applicable to the biomedical domain [22], and is now providing the advantage of common guidelines for ontology building to a multiplicity of research groups and organizations,

It is a well-documented fact that well-built, coherent ontologies tend to be hard to understand for clinical users [23]. An *is_a* hierarchy based on BFO puts kinds of processes and kinds of objects on quite distant branches. The clinician should, nevertheless, have these associations readily available on the screen. We therefore proposed that the basis for these tools should be a viewing mechanism that should reflect the terms typically appearing together in particular clinical contexts, while the full ontology was running behind the scenes. The necessary associations may be found and activated by tracking the workflows commonly used in computer applications serving clinical practice. In the following we present several specific techniques and work styles that were employed in the development of the ACGT MO.

Lassila et al. [24] categorized ontologies according to the amount of information they contain. Their classification ascribes the term “ontology” to nearly everything that is at least a finite controlled vocabulary with unambiguous interpretation of classes and term relationships and with strict hierarchical subclass relationships between classes. We disagree with this overly liberal terminological practice. Ontologies that meet more elaborate criteria, and contain a much richer internal structure were dubbed “heavyweight” and differentiated from so-called “lightweight” ontologies [25]. Among the criteria mentioned for “heavyweight ontologies” are, besides the subtype relation discussed above, also the presence of properties, value restrictions, general logical constraints, and disjoints. The ACGT MO has been designed, in this respect, to be to a heavyweight ontology. A basic principle of ontology development

is that ontologies include only classes (types, universals) but not instances (tokens). Hence the ACGT MO does not include representations of real world instances but only of universals. One of the gold standards to be followed in order to ensure a proper structure of the taxonomy of universals, is the use of a formal subtype relation and the avoidance of the informal *is_a* relations mentioned above. The subtype relation (*is_a*) is formally defined as follows: *A is_a B* if and only if all instances of *A* are also instances of *B*.

In general, we embrace the thesis that a properly constructed ontology should steer clear of a taxonomical tree that allows multiple parent classes for the same child class (i.e. one child that inherits from multiple parents). The central aim is to avoid polysemy that often results from multiple inheritances. In the ACGT MO we completely avoided multiple inheritance.

Another problematic case that can be found in a number of medical databases, terminologies and even “ontologies,” is the presence of so called *Not Otherwise Specified* (NOS) classes, e.g. “Brain Injury Not Otherwise Specified” or classes like “*UnknownX*” (“*UnknownAffiliation*”). Only recently have a number of revisions of SNOMED CT [26, 27] led to the deactivation of concepts involving the qualifier NOS such as 262686008 Brain injury NOS (disorder) and 162035000 Indigestion symptom NOS (finding). This demonstrates an increasing realist orientation in SNOMED CT. Already Cimino in his famous “Desiderata” essay [28] had counseled against the use of NOS and similar qualifiers. “Universals” of this kind do not, in fact, have any instances of their own; rather, they merely hint at a lack of data or knowledge. The alleged instances of those universals do not exhibit any shared properties, at least not necessarily. Therefore, we avoided such classes in the ACGT MO. The review of pre-existing biomedical ontologies targeting the ACGT domain led to the decision to re-use the FMA and the GO. Furthermore, some existing medical classifications and/or controlled vocabularies have been, or will be, slightly modified and added to the ontology. An example of this type is the TNM system [19].

4 Database Integration Process

The ACGT Semantic Mediation Layer (ACGT-SM) comprises a set of tools and resources that work together to serve processes of Database Integration and Semantic Mediation. The ACGT-MO is a core resource of this system, acting as Global Schema – i.e. a global framework for semantic homogenization – providing the formalization of the domain knowledge needed to support a variety of applications oriented towards clinical research and patient care. The ACGT-SM follows a Local-as-View Query Translation approach in order to cope with the problem of database integration. This means that data is not actually integrated, but is made accessible to users via a virtual repository. This repository represents the integration of the underlying databases, and the ACGT-MO acts here as database schema, providing resources for formulation of possible queries. The virtual repository has the shape of an RDF database, and the language selected for performing queries is SPARQL [29].

The ACGT-SM comprises different tools addressing different problems, such as schema level heterogeneities and instance level conflicts both at the query and data levels. These tools are designed as independent web services that collaborate in the

mediation process and are coordinated by the Semantic Mediator. The system also includes a tool devoted to aid in the process of building mappings. This mapping tool uses the ACGT-MO, and is based on a graphical visualization of its structure. The users of this mapping tool navigate the ACGT-MO and an underlying database schema in order to mark the entities that are semantically equivalent. The architecture of this system is shown in Figure 1.

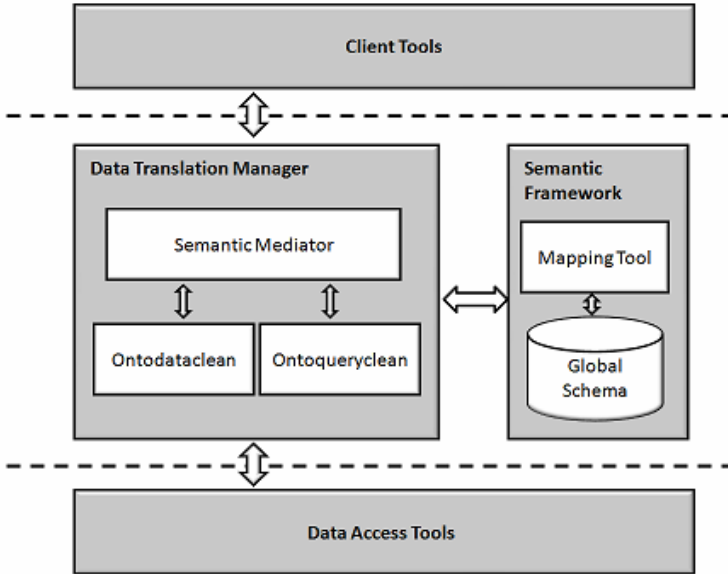


Fig. 1. ACGT Semantic Mediator Layer architecture

The ACGT-SM exposes its data services using an OGSA-DAI [30] web based interface. The OGSA-DAI middleware allows easy access and integration of data via the grid. However, no grid infrastructure is needed to access these services. The ACGT-SM offers two main services, namely 1) to launch a query, and 2) to browse the schema. The latter shows a subset of the ACGT-SM underlying RDF schema. This subset is built taking into consideration the user profile.

This software system has been tested with clinical relational and image databases [31], obtaining promising results. Currently the consortium is developing a final user query tool, with the aim of helping non technical users in the processes of building and launching queries.

5 Exploitation of the Ontology in a Clinical Trial Management System

The integration of existing data sources via the mediator is the general policy of the ACGT project. Yet the ultimate goal of ontology-based information management is to enable the direct integration of semantically consistent data created in different envi-

ronments (e.g. clinical research, laboratory data, public health data). ACGT aims to provide solutions that demonstrate the possibility of creating data in an ontology-governed way. To explore this approach, an Ontology-based Trial Management System (ObTiMA) is under development that enables those who undertake clinical trials to set up patient data management systems with comprehensive metadata by using the ACGT-MO [32]. This allows seamless integration of data collected in these systems into the ACGT mediator architecture. The main components of ObTiMA are the Trial Builder and the patient data management system. The Trial Builder allows a trial leader to define the master protocol, the Case Report Forms (CRFs) and the treatment plan for the trial in a way that is both semantically compliant with the ACGT MO and user-friendly. From these definitions, the patient data management system can be set up automatically. The data collected in the trial is stored in trial databases whose comprehensive metadata has been rendered from the start in terms of the ACGT-MO. The data can thus be seamlessly integrated through OGSA-DAI services [30] into the mediator architecture. Trial databases with comprehensive ontological metadata and the OGSA-DAI services are both automatically set up from the definitions made by the trial chairman in the Trial Builder

In the following, we briefly describe how the Trial Builder allows the clinician to define all information needed to make integration possible. In setting up a trial, clinicians want to focus on the user interfaces and to adapt them to the specific workflow of the clinical trial planned. They do not wish to be concerned with theoretical aspects and design principles of databases or ontological metadata. Therefore, in ObTiMA, the trial leader defines both, by creating the CRFs for the intended trials. He is assisted by ObTiMA in defining the questions on the CRFs, the order in which the questions will be queried, and constraints on the answer possibilities. Creating a question on the CRF is supported by simply selecting appropriate terms from the ACGT MO. For example, assuming that the clinician wants to collect all information on a patient's gender. He observes that a relation between the classes "Patient" and "Gender" exists in ACGT MO. In creating the corresponding question, he simply has to choose the class *Gender*. The attributes required in order to create the question on the CRF are then determined very easily. E.g. as answer possibilities for the question the values *Male*, *Female*, and *AmbiguousGender* are suggested, because the class *Gender* is defined as an enumeration in the ontology containing these values and a multiple choice question is subsequently automatically created on the CRF.

This procedure implements the semantics of the ontology in the CRFs in an automatic fashion.

With the aim of setting up the appropriate database for storing the data, the following attributes are needed for each question: the question itself, the data type of the answer and optionally possible data values, range constraints and measurement units. These attributes will as far as possible be determined automatically from the path the trial leader has selected, but can later be changed according to need and experience of what works best. This process leads to the possibility of lessons learned in integration of the data collected in the clinical trial at hand to be incorporated into the semantics of the ontology. In this way, the ontology itself improves in reflection of advances

made by the researchers using it. Through the integration of the ACGT-MO into ObTiMA, data sharing between clinical trials becomes possible. This is necessary to leverage the collected data for further research for example in the creation of cross-trial meta-analyses.

Ontology Viewer - Microsoft Internet Explorer

Adresse <http://obtima.ibmt.fhg.de/pages/trialdesigner/ontologyviewer.jsf>

ACGT Accelerating Clinical Oncologic Trials on Cancer

ObTiMA for clinical oncology trials

All trials My trials Query trials Create trials Administration Help

Logged in as testobtima (Logout)

Path: Patient

	Value	Exist	Count
+ hasSibling-----Sibling <input type="checkbox"/>			
hasGender-----Gender			
+ hasPerformanceRating-----PerformanceRating			
+ undergoesMedicalProcess-----MedicalProcess <input type="checkbox"/>			
+ hasPartNeoplasm-----Neoplasm <input type="checkbox"/>			
hasProgenitor-----Progenitor <input type="checkbox"/>			
hasAge-----Age			
hasWeight-----Weight			
+ hasBloodPressure-----BloodPressure			
hasMother-----Mother <input type="checkbox"/>			
hasHeight-----Height			
hasDisease-----Disease			
-----hasDisease-----NonInfectiousDisease			
-----hasDisease-----Tabagism			
+ -----hasDisease-----Syndrome			
-----hasDisease-----Alcoholism			
+ -----hasDisease-----TumorDisease			
-----hasDisease-----Neutropenia			

Fig. 2. The ObTiMA ontology viewer

We are aware that this ambitious enterprise requires tools to overcome the gap between clinical practice and biomedical reality representation. Even if an ontology provides natural language definitions for its entities and relationships (in order to make them human understandable) they are still defined in a way that is not based on practical or clinical perceptions of reality. In order to meet this desideratum, the Trial Builder provides an application-specific view on the ontology, a view that is meant to assist clinicians engaged in clinical practice or clinical trial management.

Recent studies showed that, under three different scenarios, the accuracy of SNOMED coding is only slightly over 50 % [33, 34]. One additional potential advantage of ObTiMA

is that it may help put an end to some of the problems currently faced by those using coding techniques to map clinical data unto biomedical terminologies.

6 Conclusions

The ACGT project provides a novel terminological resource for cancer research and management. It has long been recognized that an obvious application for an ontology resource is to provide a stable common schema for a mediation system such as the one that serves integration across the ACGT network. ACGT has addressed also another problem which is to provide more efficient and reliable tools for coding of clinical data by providing an ontology-driven Clinical Trial Management system which aids the clinician in collecting the data in a way compliant with the ontology.

References

- [1] Buetow, K.H.: Cyberinfrastructure: Empowering a Third Way, *Biomedical Research. Science Magazine* 308(5723), 821–824 (2005)
- [2] Tsiknakis, M., Brochhausen, M., Nabrzyski, J., Pucaski, J., Potamias, G., Desmedt, C., Kafetzopoulos, D.: A semantic grid infrastructure enabling integrated access and analysis of multilevel biomedical data in support of post-genomic clinical trials on Cancer. *IEEE Transactions on Information Technology in Biomedicine, Special issue on Bio-Grids* 12(2), 205–217 (2008)
- [3] Cali, A.: Reasoning in Data Integration Systems: Why LAV and GAV Are Siblings. In: Zhong, N., Raś, Z.W., Tsumoto, S., Suzuki, E. (eds.) *ISMIS 2003. LNCS (LNAD)*, vol. 2871, pp. 562–571. Springer, Heidelberg (2003)
- [4] <http://www.ifomis.org/acgt/1.0>
- [5] Tsiknakis, M., et al.: Building a European Biomedical Grid on Cancer, Challenges and Opportunities of HealthGrids. In: *Procs. of the HealthGrid 2006 conference, Valencia, Spain*, pp. 247–258 (2006)
- [6] <http://nciterms.nci.nih.gov/NCIBrowser/Dictionary.do>
- [7] <http://obofoundry.org/ro>
- [8] Ceusters, W., Smith, B., Goldberg, L.: A Terminological and Ontological Analysis of the NCI Thesaurus. *Methods of Information in Medicine* 44, 213–220 (2005)
- [9] <http://www.mged.org>
- [10] <http://sig.biostr.washington.edu/projects/fm>
- [11] <http://www.geneontology.org>
- [12] Smith, B., Brochhausen, M.: Establishing and Harmonizing Ontologies in an Interdisciplinary Health Care and Clinical Research Environment. In: Blobel, B., Pharow, P., Nerlich, M. (eds.) *eHealth: Combining Health Telematics, Telemedicine, Biomedical Engineering and Bioinformatics to the Edge*, pp. 219–234. IOS Press, Amsterdam (2008)
- [13] Dörr, M.: The CIDOC CRM - An Ontological Approach to Semantic Interoperability of Metadata. *AI Magazine* 24(3)
- [14] <http://protege.stanford.edu>
- [15] <http://www.w3.org/2004/OWL>
- [16] Smith, B., Ceusters, W., Klagges, B., Kohler, J., Kumar, A., Lomax, J., Mungall, C.J., Neuhaus, F., Rector, A., Rosse, C.: Relations in Biomedical Ontologies. *Genome Biology* 6, R46 (2005)

- [17] DeVita Jr., V.T., Hellman, S., Rosenberg, S.A. (eds.): *Cancer. Principles and Practice of Oncology*, 6th edn., Philadelphia. Lippincott Williams & Wilkins (2001)
- [18] Schulz, W.A.: *Molecular Biology of Human Cancers*. Springer, Dordrecht (2005)
- [19] Wittekind, C., Meyer, H.J., Bootz, F.: *TNM, Klassifikation maligner Tumoren*. Springer, Berlin (2002)
- [20] <http://suo.ieee.org>
- [21] <http://www.ifomis.org/bfo>
- [22] Grenon, P., Smith, B., Goldberg, L.: *Biodynamic Ontology: Applying BFO in the Biomedical Domain*. In: Pisanelli, D.M. (ed.) *Ontologies in Medicine*, pp. 20–38. IOS Press, Amsterdam (2004)
- [23] Rector, A.L., Zanstra, P.E., Solomon, W.D., Rogers, J.E., Baud, R., et al.: *Reconciling Users Needs and Formal Requirements: Issues in developing Re-Usable Ontology for Medicine*. *IEEE Transactions on Information Technology in BioMedicine* 2(4), 229–242 (1999)
- [24] Lassila, O., McGuinness, D.: *The role of frame-based representation on the Semantic Web*, Technical Report KSL- 01-02, Knowledge System Laboratory. Stanford University, Stanford (2001)
- [25] Gómez-Pérez, A., Fernández-López, M., Corcho, O.: *Ontological Engineering*. Springer, London (2004)
- [26] <http://www.snomed.org/snomedct>
- [27] Ceusters, W., Spackman, K.A., Smith, B.: *Would SOMED CT benefit from Realism-Based Ontology Evolution?* In: *American Medical Informatics Association 2007 Annual Symposium Proceedings, Biomedical and Health Informatics: From Foundations to Applications to Policy*, Chicago, IL, pp. 105–109 (2007)
- [28] Cimino, J.J.: *Desiderata for controlled medical vocabularies in the Twenty-First Century*. *Methods Inf. Med.* 37(4–5), 394–403 (1998)
- [29] SPARQL Query Language for RDF, <http://www.w3.org/TR/rdf-sparql-query>
- [30] The OGSADAI Project, <http://www.ogsadai.org.uk>
- [31] Martín, L., Bonsma, E., Anguita, A., Vrijnsen, J., García-Remesal, M., Crespo, J., Tsinkakis, M., Maojo, V.: *Data Access and Management in ACGT: Tools to Solve Syntactic and Semantic Heterogeneities Between Clinical and Image Databases*. In: *Advances in Conceptual Modeling – Foundations and Applications*. LNCS, pp. 24–33. Springer, Heidelberg (2007)
- [32] Weiler, G., Brochhausen, M., Graf, N., Hoppe, A., Schera, F., Kiefer, S.: *Ontology Based Data Management Systems for post-genomic clinical Trials within an European Grid Infrastructure for Cancer Research*. In: *Proc. of the 29th Annual International Conference of the IEEE EMBS*, Lyon, France, August 23-26, 2007, pp. 6434–6437 (2007)
- [33] Andrews, J.E., Richesson, R.L., Krischer, J.: *Variation of SNOMED CT coding of clinical research concepts among coding experts*. *J. Am. Med. Inform. Assoc.* 2007 14(4), 497–506 (2007)
- [34] Chiang, M.F., Hwang, J.C., Yu, A.C., Casper, D.S., Cimino, J.J., Starren, J.: *Reliability of SNOMED-CT coding by three physicians using two terminology browsers*. In: *AMIA 2006 Symposium Proceedings*, pp. 131–135 (2006)

An Ontology-Based Crawler for the Semantic Web

Felix Van de Maele¹, Peter Spyns², and Robert Meersman²

¹ Collibra nv/sa

Ransbeekstraat 230, B-1120 Brussel, Belgium

felix@collibra.com

² Semantics Technology and Applications Research Laboratory (STAR Lab)

Department of Computer Science

Vrije Universiteit Brussel

Pleinlaan 2 Gebouw G-10, B-1050 Brussel, Belgium

(peter.spyns, robert.meersman)@vub.ac.be

Abstract. We present work in progress on automated and ontology-guided discovery, extraction and mapping of information sources on the Semantic Web. It concerns an *ontology-guided focused crawler* to discover and match different data sources. We have developed an automated ontology-matcher embedded in the crawler that relates semantic web documents found during the crawl to an initial topic ontology that describes the domain of interest of the crawl. Similarity coefficients resulting from the matching process are used to guide the crawler to the information sources relevant to the modelled domain of interest. In addition, the matching process also provides links between the different data sources, which helps in the integration of this data at a later stage. In this paper, the overall architecture, the various modules and methods actually implemented are presented.

1 Introduction

Interoperability is the ability to exchange and (re)use information and has become a basic requirement in modern Information Systems. Increasing focus is put on semantics and ontologies to achieve interoperability of information systems [14]. A similar evolution can be witnessed on the World Wide Web, where semantics play an ever more important role. The ability to find, extract and match specific (semantically) annotated information constitutes the main reason for its increasing success. The Semantic Web will very likely host a lot of independently developed ontologies. These will need to be mapped and aligned to each other if one wants to realize the full potential of the Semantic Web. But before mapping and alignment can take place, we need a way to discover these different ontologies. Therefore, we have developed an ontology-based focused crawler. Our crawler, guided by an ontology describing the domain of interest, crawls the Semantic Web focusing on pages relevant to a given topic ontology. As a result, ontologies found during the crawl will be (i) relevant to the domain and (ii) produce a set of candidate mappings with the topic ontology.

The remainder of this paper is structured as follows: we give a brief overview of related work in section 2. In section 3 we elaborate on the crawler. We focus on the matching process in section 4. Section 5 contains an outline of an evaluation plan. Future work and conclusions are presented in section 6.

2 Related Work

As stated above, we define a *focused crawler* to discover semantic web data. A focused crawler is designed to only gather pages relevant to a certain, pre-defined set of topics, avoiding to explore all web pages. By definition, focused crawlers use some sort of heuristic to rate pages according to their relevance to a given topic. Focused crawlers were introduced in [4] and [7]. The typical problem for focused crawlers is to find an appropriate relevance computation function. Several improvements have been proposed since. Most of these use the link-structure of the web to improve the relevance computation. In [2], the authors examine the patterns of document-to-document correlations along web link paths to achieve more efficient crawling techniques. A recurring problem in focused crawling is finding relevant pages that are surrounded by non-relevant pages. One remedy, called *tunnelling*, is presented by Aggarwal et al. [1]: their intelligent crawler uses the characteristics of the linkage structure of the web while performing the crawl. Another method is introduced by Diligenti et al. [9] who use a context-based algorithm that builds a model for the context wherein topically relevant pages occur on the web.

One approach, by Ehrig et al. [12] is similar to ours in that an ontology is used to model the topic of the crawl. The authors provide a relevance computation method that tries to map the content of a web document against an ontology to obtain a relevance score. In our approach, we provide a similar algorithm to match the topic ontology to the text of the web pages. However, we also provide an ontology matching algorithm that maps and computes the relevance of the topic ontology to the ontologies discovered. According to our knowledge, this has not yet been presented in the current literature.

Another related approach is OntoKhoj: A Semantic Web Portal for Ontology Searching, Ranking and Classification [17]. Their goal is to provide a portal to allow agents and ontology engineers to retrieve trustworthy, authoritative knowledge and expedite the process of ontology engineering through extensive reuse of ontologies. The crawling process in Ontokhoj is however not focused and does not match the discovered ontologies to a given topic ontology.

In Ding et al. [10], Swoogle is introduced as a search and metadata engine for the semantic web. Swoogle, however, does not use a focused approach to crawl the web either and the discovered ontologies are not mapped to one to another.

3 Crawling the Semantic Web

3.1 Structure of the Semantic Web

Our view on the Semantic Web and its mutual relations is inspired by previous work from Ding et al. on the Swoogle metadata engine [11]. We introduce a vocabulary which we will use throughout this paper (based on [11]). We define Semantic Web Documents (SWDs) as documents that are freely available on the Semantic Web and are described in RDF syntax. We distinguish two different kinds of SWDs: Semantic Web Ontologies (SWOs) and Semantic Web Databases (SWDBs). Resources where the statements in the document define for the most part new concepts and relations are considered SWOs. When a document mostly introduces instances or individuals and makes assertions about them we consider it a SWDB - but this distinction is not strict.

3.2 Navigating on the Semantic Web

Hyperlinks in web documents create a meaningful way for humans to navigate on the web since we can read (i.e. interpret meaningfully) the surrounding context in a web document (assuming of course, that the hyperlink is genuine). Software agents cannot use these links in the same meaningful way. The Semantic Web endeavours to extend the web [3] by providing convenient meaningful ways to connect different data sources, at least in theory. In practice however, there are still several issues that have to be overcome. For one, interconnecting several SWOs is not a trivial task. There are ways for SWOs to link to each other (for example: *rdfs:seeAlso*, *rdfs:isDefinedBy* and *owl:imports*), but these are rarely used as ontologies usually are developed and distributed independently. Secondly, because HTML documents have no meaningful way to refer to or to embed Semantic Web documents, two parallel web universes are created. A traditional crawler engine is therefore not sufficient to find the SWD. Although likely, it is not certain the embedded SWD is relevant to a given domain of interest, even if its containing web page appears to be. Our crawler uses the given topic ontology not only to find relevant web pages, it also matches the discovered SWDs to the given topic ontology. In this way, traditional web pages help to mutually relate SWDs as these SWDs are discovered and mapped by crawling traditional web pages. The other way around, the discovered SWDs can also help to interconnect traditional web pages, a relation which might for example be used in domain-specific web portals.

3.3 The DOGMA Framework

A DOGMA¹ ontology is inspired by a classical model-theoretic perspective [19] and organises an ontology in two distinct layers (an ontology base layer and a commitment layer). This is called the principle of *double articulation* [22]. A full formalisation of DOGMA [8] as well as details on DOGMA-related research, in particular its ontology engineering methodology are provided elsewhere [23].

3.4 Crawler Architecture

Our crawler system is highly embedded into the DOGMA framework. Initially, a topic ontology is modelled using the DOGMA Studio Workbench (the STAR Lab ontology integrated development environment²). This topic ontology describes the domain of interest for the user or intelligent agent. Ideally the crawler will use this topic ontology to find only relevant web pages and discard any pages that are not relevant to the domain of interest of the agent.

The Crawl Frontier. The crawl frontier stores the URLs extracted from the web pages during the crawl. When a web page is processed and its relevance score is computed, the links extracted from that particular web page are given this relevance score. When a crawler fetches a new URL from the frontier, the URL with the highest relevance score is returned. This focuses the crawl on pages relevant to the given topic ontology. This

¹ Developing Ontology-Grounded Methods and Applications.

² <http://www.starlab.vub.ac.be/website/dogmastudio>

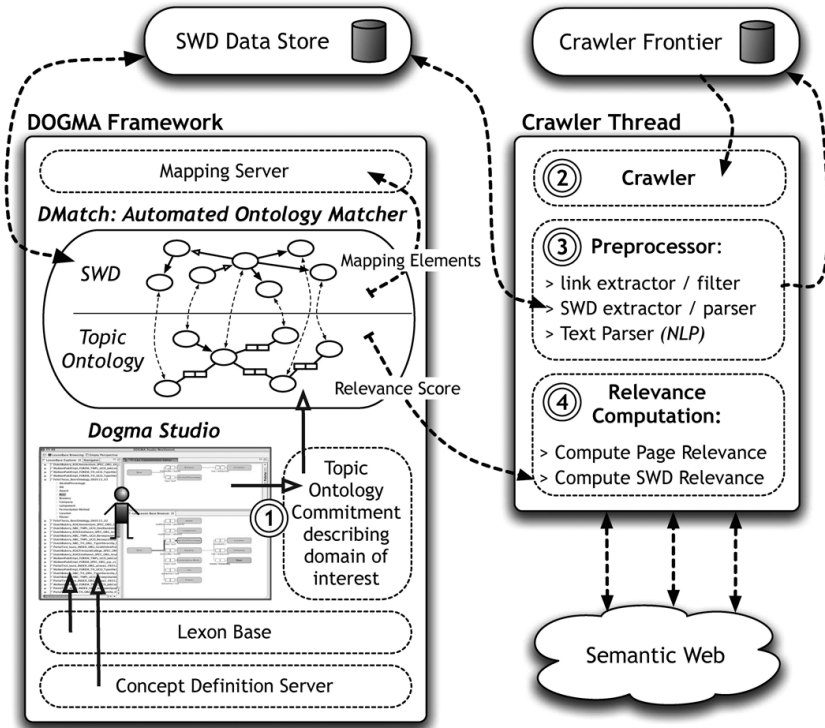


Fig. 1. The Crawler Framework

process implicitly induces a popularity factor: pages with a lot of in-links coming from pages relevant to the topic ontology will have more chance to be discovered than pages with less in-links. Therefore, “popular” SWOs have a higher chance to be discovered than rarely-used SWOs. The crawl frontier is implemented by a standard DBMS system. A direct or SOAP-based connection can be used to connect to the frontier.

The SWD Data Store. The SWD Data Store contains the SWDs that have been discovered and extracted during the crawl. For the moment, we use the Jena framework³ to store and query SWDs, but we plan to incorporate the data store in the DOGMA Studio software suite. To compute the relevance of the extracted SWDs to the domain of interest, they have to be matched to the topic ontology. For this, we have developed an ontology matcher within the DOGMA framework. The output of the matching process is twofold; (1) it will produce a similarity score that is used to compute the relevance of the SWD to the given topic ontology and (2) it will output a set of candidate mapping elements. A mapping element describes the relation and similarity coefficient between a concept from the SWD and a concept from the topic ontology. These mapping elements are stored in the Mapping Server to be reused at a later stage. As this ontology matching process is the core of our crawler framework, we provide more details in section 4.

³ <http://jena.sourceforge.net/>

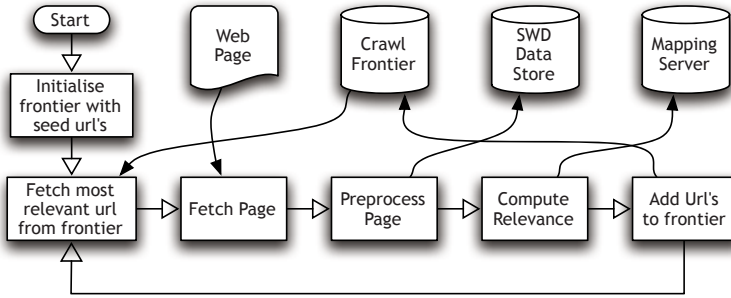


Fig. 2. The Flow Diagram of the *Crawling Loop*

The Mapping Server. The mapping server is an important part of the crawler system. All the mapping elements from the different ontologies found during the crawl are stored in the mapping server. These can be re-used to facilitate the full integration of the extracted ontologies at a later state. In further work, more services could be added to the mapping server. E.g., it could be deployed as a semantic web service and be used to query for specific ontology mappings.

The Crawler Process. The crawler itself consists of a variable number of (independent) crawler threads. Each such thread executes what is called the *crawling loop*: it will download the pages from the web, pre-process them, compute their relevance score compared to the given topic ontology and start over.

The crawler loop is started by initialising the crawl frontier with a number of URLs from where the crawl should be started. The first real step in the crawler loop is fetching the URL with the highest relevance score from the crawl frontier. After having downloaded the page into local memory, the page is pre-processed, the relevance score of the page is computed and the URLs found on the page are added to the crawl frontier. We will now discuss the pre-processing and relevance computation phases in more detail.

The Pre-processing Phase. When a page is pre-processed, all information on the page is parsed and stored in its designated data store. First, the links on the page are extracted, filtered and stored on the crawl frontier. Then, the available SWDs are extracted and added to the SWD Store. Finally, a text parser is used to analyse the text on the web page. We use well known Natural Language Processing (NLP) techniques from the Information Retrieval community that have proven their success. To compute relevant statistics of the text, we parse it into a Vector Space Model representation. We implemented four different steps to parse the text: (i) a decoder is used to strip the text from all HTML and other tags. Next, (ii) the plain text is parsed into a list of words by a tokenizer algorithm. (iii) The resulting words are filtered by a word filter. This greatly reduces the size of the vector space model. Finally, (iv) a Porter stemmer is used to strip the terms into their base form.

The Relevance Computation Phase. Our relevance computation algorithm that matches the extracted SWDs to the topic ontology is the main feature of our crawler

and distinguishes it from existing approaches. As it is of such an importance, we will discuss it in more detail in the next section. However, as the amount of (significantly) annotated web pages is still very limited, we also need to compute the relevance score of a web page when no SWD is available. In this case, we use the Vector Space Model from the pre-processing phase to compute the TF-IDF⁴ score [20] for each term corresponding with a concept from the topic ontology. We multiply these scores to get the final relevance score of the web page. The resulting score is a good representation of the relevance of the web page compared to the concepts of the topic ontology.

4 DMatch: Automated Ontology Matcher

As introduced briefly in the previous section, the relevance computation phase in the crawler loop primarily provides the crawler with a relevance score expressing the similarity between the extracted SWDs and the topic ontology. The better the concepts from each ontology match, the higher the overall relevance score will be.

4.1 The Matching Context

Many different solutions to the matching problem have been proposed - cf. [13] for a recent overview of the state of the art⁵. In this paper however, we have adopted a different focus compared to many existing approaches. As our matcher is incorporated in a crawler environment, other requirements and limitations apply. For instance, given the workflow of our crawler loop, a new page can only be fetched after the current page is fully processed. That is, when its relevance score has been computed. As a consequence, the matching process has to compute the relevance score in semi real-time. This seriously limits the number of techniques we can use for our matcher. In the following sections, we will show how we have adapted our matching process accordingly.

4.2 The Matching Methodology

Independently of the integration strategy adopted, ontology integration is divided into several methodological steps: relating the different ontologies, finding and resolving conflicts in the representation of the same world concepts and eventually merging the conformed ontologies into one global ontology [6]. We have based our matching process on this methodology.

Feature Engineering. In Figure 3 we have illustrated the methodology adopted in our matching process. The first step, the feature engineering, transforms the initial representation of both ontologies into a format that is usable for our similarity metrics. Our matcher will parse the SWO and the topic ontology, regardless in which ontology language the SWO is described, to an Ontology Model responding to a similar set of semantics. Our crawler currently supports only RDFS expressiveness. We also need a

⁴ The TF-IDF score (term frequency - inverse document frequency) is used to evaluate how relevant a word is to a document in a collection or corpus.

⁵ Or visit the following link: <http://www.ontologymatching.org/publications.html>

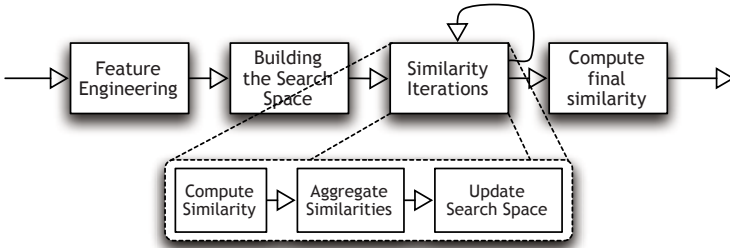


Fig. 3. The DMatch Matching Methodology

structure to compare the concepts from both ontologies. We define this structure as a *mapping element*. One rule maps one concept from the topic ontology to one concept from the SWO.

Definition 1 (Mapping Element). We define a mapping element \mathcal{M} between 2 concepts $c_i \in \mathcal{C}_\Omega$ and $c_j \in \mathcal{C}_W$ as a structure $\langle m_{id}, \gamma(\zeta, t_i), \mathcal{R}, c_j, sc(c_i, c_j) \rangle$ where:

- m_{id} stands for a mapping-id that uniquely identifies the mapping rule,
- $\gamma(\zeta, t_i)$ is the lift up of a DOGMA term $t_i \in T_i$ into a Concept $c_i \in \mathcal{C}_{\Omega_i}$ from the Dogma ontology commitment Ω ,
- \mathcal{R} is the linguistic relation between the 2 concept labels in the mapping rule,
- $c_j \in \mathcal{C}_W$ is a Semantic Web Concept from the SWO W ,
- $sc(c_i, c_j)$ is the normalised similarity score,

Building the Search Space. The derivation of the mappings takes place in a two-dimensional grid, the *search space*, consisting of all the mapping elements. During the matching process, the similarity scores of the mapping elements are updated. The search space is then used to compute the final relevance score between both ontologies. An example search space is depicted in table 1.

Table 1. An example search space

	$w_1 \in \mathcal{C}_W$	$w_2 \in \mathcal{C}_W$	$w_3 \in \mathcal{C}_W$	$w_4 \in \mathcal{C}_W$
$c_1 \in \mathcal{C}_\Omega$	$\mathcal{M}_{(c_1, w_1)}$	$\mathcal{M}_{(c_1, w_2)}$	$\mathcal{M}_{(c_1, w_3)}$	$\mathcal{M}_{(c_1, w_4)}$
$c_2 \in \mathcal{C}_\Omega$	$\mathcal{M}_{(c_2, w_1)}$	$\mathcal{M}_{(c_2, w_2)}$	$\mathcal{M}_{(c_2, w_3)}$	$\mathcal{M}_{(c_2, w_4)}$
$c_3 \in \mathcal{C}_\Omega$	$\mathcal{M}_{(c_3, w_1)}$	$\mathcal{M}_{(c_3, w_2)}$	$\mathcal{M}_{(c_3, w_3)}$	$\mathcal{M}_{(c_3, w_4)}$
$c_4 \in \mathcal{C}_\Omega$	$\mathcal{M}_{(c_4, w_1)}$	$\mathcal{M}_{(c_4, w_2)}$	$\mathcal{M}_{(c_4, w_3)}$	$\mathcal{M}_{(c_4, w_4)}$
$c_5 \in \mathcal{C}_\Omega$	$\mathcal{M}_{(c_5, w_1)}$	$\mathcal{M}_{(c_5, w_2)}$	$\mathcal{M}_{(c_5, w_3)}$	$\mathcal{M}_{(c_5, w_4)}$

Similarity Iterations. Our matching process can be classified as a *hybrid matcher* [18]: It uses several matching approaches and combines their results to efficiently compute the similarity score of each mapping element. This is done in an iterative manner where the similarity score and possibly the linguistic relation of every mapping elements is updated with each iteration. This approach allows us to easily adopt the matching process when the limitations and constraints set by the matching context change.

In our current DMatch matching system, we have implemented several matching metrics, combining different levels of matching. Initially, the terms representing the concepts are matched using well-known string-matchers such as Levenshtein distance [21] and Soundex similarity [16]. In the next step, the concepts are matched on a linguistic level, using WordNet [15]. The concepts are mapped to (several) WordNet synsets (if possible) and the linguistic relations between these synsets are then extracted. Of course, as the crawler looks for topic-specific information, domain-specific thesauri could also be used. On a semantic level, the context of the concepts are compared, which can solve the problem of homonyms. We also compare the gloss description of the concepts (if available) using the Jaccard Similarity [5]. This is also a semantically more "rich" metric than pure string-based matching - see [24] for more details.

Final Similarity Computation. In the last step, we have to compute the relevance score of both ontologies and return a list of candidate mappings. Therefore, we need an algorithm that converts the concept-to-concept mappings from the search space into one global similarity score. Since our goal is not to fully integrate the ontologies, but to compute their relevance score and to provide the framework with a set of candidate mapping elements, we have developed a greedy, sub-optimal, one-to-one mapping algorithm. It is an efficient mechanism that returns a good final similarity score [24].

5 Evaluation

The quality of a focused crawler may be evaluated on its ability to retrieve "good" pages, so in our case, "good" ontologies. As the availability of ontologies on the Semantic Web is still very limited, we will focus our evaluation of the crawler on its ability to retrieve good web pages, suspecting that discovering relevant web pages will likely yield more relevant ontologies. The main problem lies in the way good pages can be recognised. One approach is to use real users to judge the results of the crawler. This requires a high number of crawls conducted by a high number of users, which takes a lot of time. A second approach is to define a set of seed and target pages and compute the well known precision and recall metric. This method requires a well-defined and controlled crawl environment. Both approaches fall outside the scope of this paper.

6 Conclusions and Future Work

In this paper, we have proposed a framework to facilitate the automated and ontology-guided discovery, extraction and mapping of information sources on the Semantic Web. To this end, we have introduced an ontology-based focused crawler. The crawler is guided by a topic ontology in order to efficiently discover pages relevant to the domain of interest. Furthermore, to enhance the discovery, we have introduced our DMatch Ontology Matcher to match the available Semantic Web Documents to the topic ontology. This distinguishes our approach from existing crawlers.

In a preliminary analysis, we have shown that this focused approach performs much better at discovering relevant pages than a general exhaustive crawler. Combining the

extraction of domain relevant ontologies and providing the user with candidate mappings between them makes this framework very useful in a number of applications such as ontology engineering, (domain-specific) semantic web portals, semantic web services, semantic search engines, etc.

As the introduced framework consists of several different sub-systems, there is ample room to refine and further extend these subsystems. More work is needed to extend the mapping server and make it a part of the DOGMA Framework. By extensively using existing mapping elements from the mapping server, the ontology matcher might be extended to support a community-driven matching process which can further enhance the quality of the extracted semantic web documents and candidate mapping elements. The ontology matcher should also be extended to support owl-semantics. A more in-depth analysis of the ontology matcher and the performance of the crawler compared to existing crawlers was out of the scope of this paper, but is still required.

References

1. Aggarwal, C.C., Al-Garawi, F., Yu, P.S.: Intelligent crawling on the world wide web with arbitrary predicates. In: Proceedings of the 10th International World Wide Web Conference, Hong Kong, pp. 96–105 (May 2001)
2. Bergmark, D., Lagoze, C., Sbityakov, A.: Focused crawls, tunneling, and digital libraries. In: Agosti, M., Thanos, C. (eds.) ECDL 2002. LNCS, vol. 2458, pp. 91–106. Springer, Heidelberg (2002)
3. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Scientific American* 284(5), 34–43 (2001)
4. Chakrabarti, S., van den Berg, M., Dom, B.: Focused crawling: a new approach to topic-specific web resource discovery. In: WWW 1999: Proceeding of the eighth international conference on World Wide Web, pp. 1623–1640. Elsevier North-Holland, Inc. (1999)
5. Cohen, W., Ravikumar, P., Fienberg, S.: A comparison of string distance metrics for name-matching tasks. In: Proceedings of the IWeb Workshop at the 22 IJCAI conference (2003)
6. De Bo, J., Spyns, P., Meersman, R.: Towards a methodology for semi-automatic ontology aligning and merging. Technical report 02, Vrije Universiteit Brussel - STAR Lab, Brussel (2004)
7. De Bra, P., Post, R.: Information retrieval in the World-Wide Web: Making client-based searching feasible. *Computer Networks and ISDN Systems* 27(2), 183–192 (1994)
8. De Leenheer, P., Meersman, R.: Towards a formal foundation of dogma ontology: part i. Technical Report STAR-2005-06, VUB STAR Lab, Brussel (2005)
9. Diligenti, M., Coetsee, F., Lawrence, S., Giles, C.L., Gori, M.: Focused crawling using context graphs. In: 26th International Conference on Very Large Databases, VLDB 2000, pp. 527–534. Morgan Kaufmann, San Francisco (2000)
10. Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V.C., Sachs, J.: Swoogle: A Search and Metadata Engine for the Semantic Web. In: Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management. ACM Press, New York (2004)
11. Ding, L., Pan, R., Finin, T., Joshi, A., Peng, Y., Kolari, P.: Finding and ranking knowledge on the semantic web. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 156–170. Springer, Heidelberg (2005)
12. Ehrig, M., Maedche, A.: Ontology-focused crawling of web documents. In: Proc. of the 2003 ACM symposium on Applied computing, Melbourne, Florida (2003)

13. Euzenat, J., Shvaiko, P.: *Ontology matching*. Springer, Heidelberg (2007)
14. Euzenat, J.: Towards a principled approach to semantic interoperability. In: Gómez-Pérez, A., Gruninger, M., Stueckenschmidt, H. (eds.) *Proceedings IJCAI 2001 Workshop on ontology and information sharing*, pp. 19–25 (2001)
15. Miller, G.: Wordnet: a lexical database for english. *Comm. ACM* 38(11), 39–41 (1995)
16. Mortimer, J.Y., Salathiel, J.A.: Soundex codes of surnames provide confidentiality and accuracy in a national hiv database. *CDR Rev.* (1995)
17. Patel, C., Supekar, K., Lee, Y., Park, E.K.: Ontokhoj: a semantic web portal for ontology searching, ranking and classification. In: *WIDM 2003: Proceedings of the 5th ACM international workshop on Web information and data management*, pp. 58–61. ACM Press, New York (2003)
18. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. *The VLDB Journal* 10(4), 334–350 (2001)
19. Reiter, R.: Towards a logical reconstruction of relational database theory. In: Brodie, M., Mylopoulos, J., Schmidt, J. (eds.) *On Conceptual Modelling*, pp. 191–233 (1984)
20. Salton, G., McGill, M.J.: *Introduction to modern information retrieval*. McGraw-Hill, New York (1983)
21. Sankoff, D., Kruskal, J.B. (eds.): *Time warps, string edits, and macromolecules: The theory and practice of sequence comparison*. Addison-Wesley Publication, Reading (1983)
22. Spyns, P., Meersman, R., Jarrar, M.: Data modelling versus ontology engineering. *SIGMOD Record Special Issue* 31(4), 12–17 (2002)
23. Spyns, P., Tang, Y., Meersman, R.: An ontology engineering methodology for DOGMA. *Journal of Applied Ontology* 5 (in print, 2008)
24. Van de Maele, F.: *Ontology-based crawler for the semantic web*. Master's thesis, Vrije Universiteit Brussel - STAR Lab, Belgium (2006)

Consensus Emergence from Naming Games in *Representative Agent* Semantic Overlay Networks

Gabriele Gianini, Ernesto Damiani, and Paolo Ceravolo

Università degli Studi di Milano, Dipartimento di Tecnologie dell'Informazione
via Bramante 65, 26013, Crema - Italy
{gianini,damiani,ceravolo}@dti.unimi.it

Abstract. Language, as a shared set of conventions for mapping meanings to expressions, can emerge from the self-organization - into a global consensus state - of a population of distributed agents connected through some communication network and playing local collaborative games such as the Naming Game. Concepts and methods involved in this problem are very similar to those applied in statistical physics. In this work we propose a kind of self-organizing Semantic Overlay Networks, inspired by the mechanics of the Ising spin model - and undergoing a variant of distributed simulated annealing - which can converge to a consensus vocabulary through the abrupt transition from disorder to order; the condition which grants the convergence (the mean-field condition - a.k.a. Representative Agent condition - of everyone knowing about the state of everybody else) is approximated here by a sampling, performed through a suitably randomized message exchange mechanism. We outline two possible implementation of such kind of networks: one based on a structured, the other based on an unstructured P2P network.

1 Introduction

The recent appearance of peer-to-peer (P2P) applications has made possible a different approach to the issue of inter-operability at semantic level, opening a new perspective based on the social dynamics of crowds of peer agents and on a class of mechanisms of self-organization inspired to physical, biological and social systems, and studied diffusively in computer science: the different mechanisms could lead to an emergent semantics, whereby one derives a global semantic agreement, or obtains a global semantic inter-operability, from interactions and agreements stipulated locally.

In this paper we outline one of such possible distributed architectures, a Semantic Overlay Network, focusing on a simplified, core element of the semantic consensus problem: the association of a specific name to a single object. The network of agents will search for the global consensus by stipulating only local agreements in a randomized naming game where the inner working of the system will be based on a variant of a distributed simulation annealing, of which we consider two implementations: one based on a structured P2P networks (a distributed index network, with query forwarding), the other based on an unstructured

network (working through gossiping). Several analogies between social interactions among peers and statistical physics will be employed in the specification of the complex system behavior. In complex systems, global behaviors can emerge from local interactions between system entities: the emergence phenomenon is characterized by an abrupt transition in the phase (quantity or quality) of a global system property, the so-called order variable, in correspondence to the critical level of one or more control parameters defining the local interactions. This sudden transition is known as phase transition. In the paper we will deal with two examples of phase transitions, one inspired by statistical physics – more specifically by Ising spin systems – the other by percolation theory. In the former the order parameter is the correlation (i.e. the agreement) over possible configurations in a configuration space (representing all the possible mappings from symbol to meaning): in this case we are interested in understanding the conditions of the local interaction by which the system will undergo a disorder-order phase transition, bringing the agents to a global consensus. In the latter the order parameter is the size of connected clusters (subsets of the network where nodes are mutually reachable): by varying the probability by which a message is forwarded at each node one can vary the effective size of connected clusters, and we are interested into the value by which there is a transition from a partitioned communication topology to a fully connected communication topology. This mechanism – the so called probabilistic flooding [4] – will apply to the unstructured network implementation of the system.

The paper is organized as follows: first we review the most relevant mechanisms used to reach a global consensus based on local interactions (Section 2), ranging from voter dynamics to Ising model inspired dynamics (simulated annealing, Glauber dynamics) to the more rich and specific Naming Games' dynamics; then we describe our Semantic Overlay Networks (Section 3): the first based on a simplified version of Chord, the other based on a simplified version of Gnutella; a discussion of the difference with respect to the related work (Section 4) and an outline of the future work concludes the paper.

2 Distributed Consensus Mechanisms

There is a rich variety of different mechanics able to lead a distributed set of agents to a global consensus through the only use local interactions. We examine here the main ideas, describing and discussing the relation among the different dynamics, with particular reference to a kind of Voter Dynamics – the Ising spin model with Glauber-Metropolis Dynamics – and to the Naming Game Dynamics.

All the processes considered are continuous time processes where the next state is obtained by some update rule (the dynamics of the process), either deterministically or stochastically, based on the current state. In the Ising spin model, we consider processes with a binary state space: there is an individual at each site $x \in S$ (equivalently, network node), who has possible opinions -1 and $+1$ at any given time (a spin with two possible states) [8]. In the Naming Game Model instead the state of an agent is defined by the repertoire of names he

knows of a given object. In all the cases the dynamics of the process is specified by a set of *transition probabilities* between different states: for a voter model the probability by which at a given site there is a flip from -1 to $+1$ or viceversa is given by a function of the site x and the configuration η of the system and of some noise level T ; when $T = 0$ one recovers a deterministic rule. The evolution of the consensus in the system can be tracked by means of the correlation between spins. We are interested in the existence of stationary configurations (or almost-stationary) where all the individuals – but possibly few – are in the same state.

2.1 The Glauber Dynamics for the Ising Model

The Ising Model and the Mean-Field Approximation. In the Ising model the set of spins located at the nodes interacts with *all the nearest neighbors* and has the propensity to align along the direction of the majority of the neighbors. This model, where the intensity of the interaction depends on the distance is representative of many physical systems, for instance it can be used to represent a ferromagnetic system undergoing a phase transition from disorder to order as the temperature decreases: a regular lattice of spins at high temperature (high average kinetic energy) does not show a net magnetization because – due to the kinetic fluctuations – the spins are oriented uniformly randomly over the two available states; however as the temperature decreases the spins start to take into account the effect of the next neighbors aligning along their direction; starting from a characteristic temperature, the critical temperature, the overall system starts displaying a net magnetization and eventually settles into one of the two global minimum equivalent states (all the spins along -1 , or all the spins along $+1$), i.e. it reaches a global consensus. Notice that the temperature here plays a key role: the random fluctuations introduced by a non-zero temperature can allow a spin to align, despite its own propensity to conform to the neighborhood, also on the opposite direction with respect to the majority. The system is analytically tractable only in few cases, e.g. in 1D and 2D regular cubic lattices, however an analytically tractable approximation schema is often used which for $d > 1$ gives qualitatively correct results, the so called *Mean-Field Approximation* also known in economical models as the *representative agent* approximation. In this approximation one considers, as the system cools, the behavior of a single representative spin, immersed in the field of all the other spins of the system, as if the individual spin had the possibility to interact with the whole set of spin of the system and behaved as if it felt the average of the field produced by the other spins. This simplified model produces always a disorder-order phase transition at finite temperatures. An obvious consequence of this fact is that if we can force a system into an everybody-talks-to-everybody mechanics the system will display a disorder order phase transition. We will return on this observation in the next section.

In a generalization of the Ising model, the *Potts* model each spin can assume one of m values, also in this model equal nearest neighbor values are energetically favored: the Ising model correspond to the special case $m = 2$.

Glauber Dynamics for the Ising Model. If we consider a collection of N spins (agents) s_i that can assume two values ± 1 , each energetically lean to be aligned with the set of its nearest neighbors, then the total energy of the system is $E = -\sum_{(i,j)} s_i s_j$ where the sum runs on the pairs of nearest neighbor spins. The most common type of dynamics (updating rules for the state of the system) used to make the system evolve is the so called Glauber-Metropolis' (a special case of a distributed Simulated Annealing algorithm). The Metropolis dynamics consists in an asynchronous move whereby a single spin is chosen at random and then is flipped, if the change in energy ΔE is negative the move is accepted, otherwise it is accepted with some probability decreasing with the system temperature T – the Metropolis' choice for that probability was the Boltzmann function $\exp(-\Delta E/k_B T)$ where k_B is a constant – so that that when the temperature reaches the zero the update rule changes from stochastic to deterministic. In the Glauber dynamics also good moves (negative ΔE) can be discarded with some probability, the probability of changing state is symmetrically represented by the sigmoidal function $1/(1 + \exp(-\Delta E/k_B T))$ – in the limit of zero temperature the sigmoid becomes a 0-1 step function, recovering determinism. In a fully distributed version of the Glauber dynamics each agent should alternatively send its status to the neighbors and collect the values sent by the neighbors to change its state. In many systems, once a suitable cooling schedule is followed, the propensity of each spin towards being in the same state as the neighbors' drives the system towards one of the two possible completely ordered states – with all positive or all negative orientation – as the system cools. In some system with smooth energy landscapes the ordering appears naturally at zero temperature. However in some other systems the specific topologies can create an energy landscape that hinders the convergence: in those cases a suitable level of noise, controlled by a cooling schedule, is useful to not get stuck in local minima of the energy landscape. However, in a fully connected network the convergence is granted also at zero temperature.

2.2 Naming Games

The Naming Game [11] was expressly conceived to explore the role of self organization in the evolution of language and possibly represents the simplest example of the complex processes leading progressively to the establishment of complex human-like languages. It focuses on the formation of vocabularies: each agent develops its own vocabulary in a random private fashion, however agents are forced to align their vocabularies, through successive interactions (conversations), in order to obtain the benefit of cooperating through communication. A globally shared vocabulary should emerge, as a result of local adjustments of individual word-meaning association. Notice that conversations can include also non-linguistic behavior, such as pointing.

The Minimal Naming Game. The simplest version of the Naming Game [5] is played by a population of N agents trying to bootstrap a common vocabulary

for a certain number of individual objects present in their environment, so that one agent can draw the attention of another one to an object, e.g. to obtain it or converse further about it. However if the number of possible words is so large that the probability that two players invent the same word at two different times for two different objects is practically negligible (no homonymy allowed) one can reduce the environment to one single object without loss of generality. Each player is characterized by an inventory of word-object associations he knows. All agents have empty inventories at time zero; at each time step two players are picked at random and one of them plays as speaker and the other as hearer. Their interaction obeys the following rules: (1) The speaker selects an object from the current context; (2) The speaker retrieves a word from its inventory associated with the chosen object, or, if its inventory is empty, invents a new word; (3) The speaker transmits the selected word to the hearer; (4) If the hearer has the word named by the speaker in its inventory and that word is associated to the object chosen by the speaker, the interaction is a success and both players maintain in their inventories only the winning word, deleting all the others; (5) If the hearer does not have the word named by the speaker in its inventory, or the word is associated to a different object, the interaction is a failure and the hearer updates its inventory by adding an association between the new word and the object. The game is played on a fully connected network. One can distinguish three phases in the behavior of the system. Very early, pairs of agents play almost uncorrelated games and the number of words q hence increases linearly with the time t . In the second phase the success probability is still very small and agents' inventories start getting correlated, the $q(t)$ curve presenting a well identified peak. The process evolves with an abrupt increase in the number of successful conversations and a further reduction in the numbers of words. Finally, the dynamics ends when all agents have the same unique word: the system spontaneously selects one of the many possible ordered states.

A Naming Game Equivalent to an Ising Model. Consider a naming game with a fixed word length indicated by ℓ , where an agent, upon observing the words – used to refer to an object – by a sample of agents, has the propensity to adopt a new word which is a mix of the observed words, composed by the most common letters at each position. This game would share both the feature of agents retaining some memory of past encounters, as in a Minimal Naming Game, and the feature of a majority vote between peers, akin to the (zero temperature) Ising model. If we adopt the rule of taking a decision contrasting this optimum choice with some probability according to a Glauber criterion, we have a naming game which is equivalent to a set of ℓ independent m states Potts models, where m is the number of letters allowed in the alphabet, including the separation space. With little loss of generality we can consider only words consisting in ℓ *binary* digits, in that case, since we do not allow the letters at different position to interact to one another, the game will be equivalent to a set of parallel Ising games played by the same network of nodes: we can expect the convergence to an ordered consensus state – where every agent uses the same word – under the same conditions stated above.

3 Representative Agent Semantic Overlay Networks

In the previous section we have seen that a mean-field/representative-agent Ising model – where everybody feels the effect of everybody’s opinion – even when starting from a completely random distribution of the individuals between the two possible opinions, will settle into one of the two global agreement states, after undergoing a disorder-order phase transition. An obvious consequent consideration is that, if we can design a system that through some mechanics becomes equivale: by having peer agents play the Naming Game described above, inspired to the Ising model or to the Potts model one will obtain a Semantic Overlay Network able to reach global agreements over the naming of objects (we can focus on a single object visible to everyone, either because everyone has a copy of it or a reference to it or because the copy or the reference are passed around: we disregard in this work the issue of how the nodes know the object).

A communication network is characterized naturally by two distinct topologies: the *physical topology* where two nodes are connected if they are next-neighbors and the *communication topology* where two nodes are connected if the routing mechanics can create a path from one to the other.

A situation where everyone feels the effects of everyone else can be created by making the *communication topology* fully connected. However the cost in terms message exchange for this condition, which is already expensive when the underlying physical topology is fully connected (here the cost per communication is conventionally equal to 1 and the condition costs $O(N^2)$), for a generic topology can be unaffordable. One important observation, however, is that the full communication connectivity can be substituted by a sampled communication connectivity over the whole network without losing the mean-field condition [7]: if each peer plays with only a limited number ϵ of temporary players who are chosen randomly in every step – so that there is no correlation between partners from one communication round to the next – the mean field condition is preserved. A uniform randomization will be easier and relatively cheap to obtain in a structured P2P network – where typically the cost per communication is proportional to $\log_2(N)$), whereas is generally more critical in unstructured P2P networks: in those networks one can to reduce the communication cost and the sampled fraction at the same time by tuning the probability of forwarding a message down to the level of node sampling deemed strictly necessary [4,9].

Hereafter we outline two different implementation of this class of Semantic Overlay Network, which being based on the mean-filed/representative agent condition are called *Representative Agent Semantic Overlay Networks*: the first example is based on a structured P2P communication infrastructure, the second on an unstructured communication infrastructure [6]. The two implementations although very different share a number of common traits. Each node is always playing one of few basic scenarios: a node joins the network, a node leaves the network, a node sends its status to a group of other nodes, a node receives the status of other nodes and processes the corresponding information to take a decision about its own new state, the node carries on the task of forwarding a

message and possibly updates some variable which is functional to the overall operation of the network (e.g. the noise/perturbation level for decisions).

3.1 A Structured SON

Structured P2P networks based on Distributed Hash Tables differ in the distinct costs of structural and communication operations, typically measured in routing hops [6]. We chose to use Chord as a reference because the routing of a message to a destination goes as $O(\frac{1}{2} \log_2(N))$, whereas the higher cost for joining and leaving the network – which goes as $O(\frac{1}{2} \log_2^2(N))$ – is not a big problem in a Semantic Overlay Network, where churning is expected to be low. The keys of the Chord DHT are l -bit identifiers, i.e., integers in the range $[0, 2^l - 1]$. They form a one-dimensional identifier circle modulo 2^l wrapping around from $2^l - 1$ to 0. Each node is associated to an identifier (also each data item is associated to an identifier and this is used to allocate the responsibility of each node, however, as anticipated we do not need this feature). Each node maintains a routing table, the finger table, pointing to other nodes on the identifier circle, corresponding to the first available keys separated from the node by the powers of two. Given a circle with l -bit identifiers, a finger table has a maximum of l entries. Given the power-of-two intervals of finger IDs, each hop covers at least half of the remaining distance on the identifier circle between the current node and the target identifier. This results in an average of $O(\log_2(N))$ routing hops for a Chord circle with N participating nodes. The essential features of the protocol are the following: (1) Upon Arrival a new node contacts an existing node, which based on the newcomer's id tells the other relevant nodes and communicates to the newcomer the finger tables, the number of nodes currently in the net, and possibly open naming challenges; if a naming challenge is open the newcomer node chooses its state randomly. (2) The peer generates at random a number of ids taking into account the coverage of the name space provided by the current number of active users and sends out information about its own state by using the entry in the finger table closest to the target peer. (3) The peer receives messages about the state of other peers from the predecessors, stores the message and when is not the intended target it forwards the message by using the closest entry in the finger table; when he is the closest entry and he knows that the key does not correspond to any existing peer it simply drops the message. (4) The peer periodically takes a decision about his own state based on the set of messages to which was the intended target and based on a random subset of the messages seen during a round (the round can be defined by the collection of a sufficient number of sample messages). If the naming game is a simple coordination game (analog to the ferromagnetic Ising system described above) this kind of mechanics is bound to converge to an ordering even at zero temperature. In this case there is no need for a cooling schedule.

3.2 An Unstructured SON Based on Probabilistic Flooding

The other Representative Agent Overlay we consider is an unstructured overlay, operating through gossiping: we will consider a case where there is no TTL,

but the message most of the times is passed on to a neighbor whereas it gets dropped with probability q : this rule is called *probabilistic flooding*. Flooding is an effective mechanism for both broadcast and uni-cast modes of communication, providing broad coverage and guaranteeing minimum delay, however in general is not scalable. There have been the numerous attempts to improve the scalability of unstructured P2P networks [10], a relevant one using probabilistic flooding is based on a percolation model [4]. In a typical percolation model, lines, or bonds, are drawn between sites at one hop distance from each other and each line can be opened with probability p (the two sites are connected with probability p); a cluster (or component) is defined as a group of sites among which exist communication paths; one says that a cluster *percolates* the lattice if it extends two opposite sides of a lattice. As p increases, the emergence of the giant cluster for the first time marks the critical point p_c of the phase transition. For a random graph of size N (ideally tending to infinity) with potential connectivity distribution $P(k)$, percolation theory says that criticality is reached when the ratio, we call α , of the second to first moment of the graph is exactly equal to two. This picture can be adapted to networks with probabilistic flooding and with connectivity distribution $P(k)$ by choosing an appropriate value of probability for message dropping. In real cases a distributed bookkeeping of the relevant quantities (the first moments of $P(k)$) can be performed during normal network operation: for instance if we assume there is no reason why in different areas of the network there should be a substantially different connectivity distribution, each node could periodically estimate number of its first-level to third-level neighbors' connectivity with local ping packets of limited TTL.

Hereafter we provide an example of the interaction between nodes for one of the possible variants of the distributed protocol. We make the assumption that although the network may contain a very high number of nodes, the diameter of the network (i.e. the maximum of the minimum number of hops separating any two nodes) is always reasonably limited, as it happens to the Internet (where the average minimum separation between any pair of nodes is estimated to be less than about 20 hops). The basic steps are the following: (1) At startup the client contacts a few other nodes it knows about and generates a unique node id based on a hash of its Internet address and a time stamp; the new node receives from each of the few contacted nodes an estimate of the relevant statistical parameters of the network: knowledge of the estimated size of the network and connectivity moments will be used to tune the forwarding probability and to decide how long to wait for a sampling round; furthermore, if it is the case, the new peer receives the information about (the remaining segment of) the cooling schedule; the peer takes a decision about its own state at random. (2) The peer sends out a message to all his next neighbors, advertising his own state and the local temperature: the message is bound to realize a random walk through the network until is discarded. (3) Upon reception of the message a peer appends its own state and temperature to the list, then routes the message by considering all the nearest neighbors and for each one sending the message with some probability; the information read from the message (reporting a list

of nodes, their states and the number of hops) is stored locally. (4) From time to time, having collected sufficient number of peer states, generated at a given temperature (messages generated at temperatures too high with respect to the current one are discarded), the peer performs an update based on the Glauber update rule; on the decision will impact all the information collected from the messages routed by the peer, weighted by number of hops distance (to get a balanced view of the network state a node has to take into account that messages coming from faraway nodes are less likely to be received).

A last aspect concerns the use of a finite temperature and a cooling schedule: whereas with a structured overlay one can always guarantee a uniform sampling, in unstructured overlay networks with irregular topologies may not provide such guarantee. For instance if in a network there are two distinct large well locally connected domains linked to one another by a bridge (a link whose failure makes the network partitioned in two non communicating components), it may happen that due to the difficult of reaching the mean-field informative state, two parts of the network settle in two different ordered states: this situation represents to the overall system a deep local minimum, difficult to unsettle (the information which can flow from one domain into the other is not sufficient to tilt it). With these networks one can use some suitably slow cooling schedule starting at some appropriate finite temperature.

4 Discussion and Conclusions

Several papers deal with different parts of the topics proposed in this work. The paper [3] provides a proof of concept that language games can be an effective solution to creating and managing a distributed process of agreement on a shared lexicon and describe a fully distributed service oriented architecture for language games, which implies the use of a fully connected (small) set of nodes. The work [2] addresses the formation of new concepts and their corresponding ontology in a multi-agent system where individual autonomous agents try to learn new concepts by consulting several other agents. In this research individual agents create and learn their distinct conceptualization and rather than a commitment to a common ontology they use their own ontologies. The paper [12] proposes a domain independent method for learning a mapping between ontologies, based on exchanging instances of concepts that are defined in the ontologies. The method evaluates the mappings between the ontologies using the pairs of instances: for each step of this method, the likelihood that a decision is correct is taken into account. The method implies a fully connected network. The authors of [1] paper describes a gossiping based approach for obtaining semantic inter-operability among data sources in a bottom-up, semi-automatic manner without relying on pre-existing, global semantic models, but relying on data organized and annotated according to local schemas: here participants do not try to agree over a set of global statements but provide translations between schemas they are interested in. None of the above mentioned papers tries to solve the global consensus problem by dealing with the problem of granting the

convergence. Here we recall that the mean field condition is a sufficient condition for convergence and try to approximate the condition over real networks. The paper proposes a class of self-organizing Semantic Overlay Networks, inspired by the mechanics of the Ising spin model which can converge to a consensus vocabulary by a disorder-order phase transition, corresponding to the adoption of a common vocabulary; the condition which grants the convergence is approximated here by an equivalent sampling, performed through a suitably randomized message exchange mechanism. We outlined two possible implementation of such class of networks: the two overlays differ in that one is based on a structured P2P network, and by construction can benefit of efficient communication, at the price of some infrastructure maintenance, the other is based on an unstructured network and can be made efficient by tuning the parameters of a probabilistic flooding algorithm.

References

1. Aberer, K., Cudré-Mauroux, P., Hauswirth, M.: The chatty web: emergent semantics through gossiping. In: WWW 2003: Proceedings of the 12th Int.Conf. on World Wide Web, pp. 197–206. ACM, New York (2003)
2. Afsharchi, M., Far, B.H.: Automated ontology evolution in a multi-agent system. In: InfoScale 2006: Proceedings of the 1st international conference on Scalable information systems, p. 16. ACM, New York (2006)
3. Avesani, P., Cova, M.: Shared lexicon for distributed annotations on the web. In: WWW 2005: Proceedings of the 14th international conference on World Wide Web, pp. 207–214. ACM, New York (2005)
4. Banaei-Kashani, F., Shahabi, C.: Criticality-based analysis and design of unstructured peer-to-peer networks as "complex systems". In: CCGRID 2003: Proceedings of the 3rd International Symposium on Cluster Computing and the Grid, Washington, DC, USA, p. 351. IEEE Computer Society, Los Alamitos (2003)
5. Baronchelli, A., Dall'Asta, L., Barrat, A., Loreto, V.: Topology induced coarsening in language games. *Physical Review E* 73, 015102 (2006)
6. Gotz, S., Rieche, S., Wehrle, K.: Selected DHT Algorithms. In: Steinmetz, R., Wehrle, K. (eds.) *Peer-to-Peer Systems and Applications*, ch. 8, vol. 3485, pp. 95–117. Springer, Heidelberg (2005)
7. Szabo, G., Fath, G.: Evolutionary games on graphs. *Phys. Rep.* 446, 97–216 (2007)
8. Liggett, T.M.: *Stochastic Interacting Systems*. Springer, Heidelberg (1999)
9. Menaşcé, D.A., Kanchanapalli, L.: Probabilistic scalable p2p resource location services. *SIGMETRICS Perform. Eval. Rev.* 30(2), 48–58 (2002)
10. Meshkova, E., Riihijärvi, J., Petrova, M., Mähönen, P.: A survey on resource discovery mechanisms. *Comput. Netw.* 52(11), 2097–2128 (2008)
11. Steels, L.: Self-organizing vocabularies. In: *Artificial Life V*, pp. 179–184. MIT Press, Cambridge (1996)
12. Wiesman, F., Roos, N.: Domain independent learning of ontology mappings. In: AAMAS 2004: 3rd Intern. Joint Conf. on Autonomous Agents and Multiagent Systems, Washington, pp. 846–853. IEEE Computer Society, Los Alamitos (2005)

A Semantic Crawler Based on an Extended CBR Algorithm

Hai Dong, Farookh Khadeer Hussain, and Elizabeth Chang

Digital Ecosystems and Business Intelligence Institute,
Curtin University of Technology,
GPO Box U1987 Perth, Western Australia 6845,
Australia

{hai.dong, farookh.hussain, elizabeth.chang}@cbs.curtin.edu.au

Abstract. A semantic (web) crawler refers to a series of web crawlers designed for harvesting semantic web content. This paper presents the framework of a semantic crawler that can abstract metadata from online webpages and cluster the metadata by associating them with ontological concepts. The clustering is based on a CBR algorithm which is adopted in the field of problem solving. We reveal the technical details with regard to ontological concept and metadata format, and the extended CBR algorithm. In addition, the system implementation and evaluation details are provided in detail, finalized by our conclusion and further works.

Keywords: semantic crawler, metadata abstraction, ontological concepts, extended CBR algorithm.

1 Introduction

A semantic (web) crawler refers to a series of web crawlers designed for harvesting semantic web content [4]. The semantic web content is normally marked by the ontology mark-up languages (e.g. RDF, XML, OWL, and so forth). In this paper, we will present the conceptual model of a semantic crawler. This semantic crawler uses an extended CBR algorithm to associate the harvested metadata with ontological concepts, in order to realize the purpose of metadata clustering.

The rest of the paper is organized as follows: first of all, the system architecture of the whole crawling system is presented; then we provide the ontological concept and metadata format in the OWL format, followed by the introduction of the core part of the semantic crawler – a CBR algorithm and its extended version in the field of semantic information retrieval; following that, we reveal the implementation details; next, to evaluate the performance of the semantic crawler, we choose a benchmark from the traditional information retrieval evaluation methods, and present its variations and purposes in our testing environment; the testing results are then presented and analysed; the conclusion and further works are summarized in the final section.

2 System Architecture

The design of crawling system has two primary objectives as below:

- Generating metadata by extracting structured and meaningful information from downloaded webpages.
- Clustering metadata by adding references to ontological concepts.

The whole system consists of two main parts – a semantic crawler and a knowledge base, which will be introduced in detail in the following paragraphs (Fig. 1).

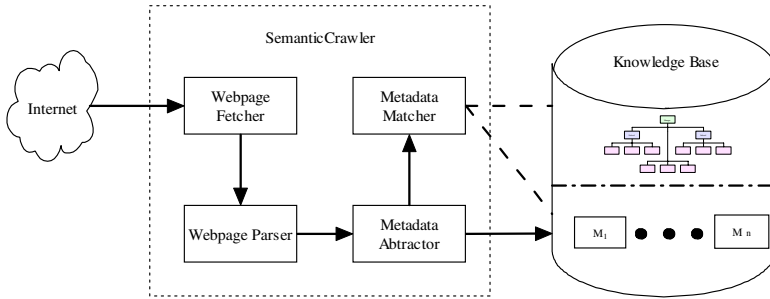


Fig. 1. System architecture

The semantic crawler architecture is composed of four agents, which are a Webpage Fetcher, a Webpage Parser, a Metadata Abtractor, and a Metadata Matcher. The function of each agent is described as follows:

- The Webpage Fetcher's mission is to selectively download webpages in a given website, by configuring the visiting URLs. Once the URL of a given website is passed to the Webpage Fetcher, it will visit and download the webpage located by the URL, and then analyze all hyperlinks, in order to choose further visiting webpages. This can be realized by setting up a set of webpage fetching rules for analyzing HTML hyperlink tags and their annotations (e.g., "next page", numbers, and so forth).
- The Webpage Parser's function is to parse the web documents in the downloaded webpages, and to filter meaningless information (e.g., hyperlinks, footnotes, unimportant HTML tags, and so on) in the parsed documents. Similar to the Webpage Fetcher, a set of webpage parsing rules are set up. According to the rules, the agent can parse the web documents based on the regulated HTML tags.
- The Metadata Abtractor's goal is to abstract the structured and meaningful information from the parsed web documents, and to form metadata based on these information. Similarly, the rules of information abstraction need to be configured, with the purpose of analyzing information from the downloaded webpages. These rules configure the HTML tags and annotations whose contexts may contain meaningful information. Based on the rules, the agent can extract the meaningful information, and treat them as the properties of metadata. By adding OWL tags, the metadata can be formed.

- The Metadata Matcher is used to cluster the metadata with predefined ontological concepts, by means of associating the semantically similar metadata with concepts. The semantic similarity between metadata and concepts are determined by an Extended CBR algorithm. The associating process is realized by inserting the URI of each side into a property of the other.

The knowledge base is used to store the predefined ontological concepts and metadata. The format of ontological concepts and metadata will be described in the next section.

Thus, the workflow of the whole system can be concluded as follows:

- First of all, the Webpage Fetcher will download a webpage according to a given URL. By means of the predefined webpage fetching rules, it will then extract the useful URLs from the webpage for the further visit. The above is a recursive process.
- For the downloaded webpages, the Webpage Parser will obtain all web documents from them, and filter the meaningless information according to the webpage parsing rules. The parsed documents will be passed to the Metadata Abstractor.
- After receiving the parsed documents, the Metadata Abstractor will extract the meaningful information upon the metadata abstracting rules, then form metadata by using these information. The metadata will then be stored into the knowledge base.
- Once a metadata is stored into the knowledge base, the Metadata Matcher will use the Extended CBR algorithm to determine the semantic similarity between the metadata and all ontological concepts in the knowledge base. If the metadata and a concept is similar, the URI of each side is then stored into the property of the other side. Therefore, the metadata are associated and clustered with ontological concepts.

3 Ontological Concept and Metadata Format

In this section, the format of ontological concepts and metadata will be represented in the form of OWL.

3.1 Ontological Concept Format

Each ontological concept has two basic properties (the number of properties can be extended according to the real environment), which are `conceptDescription` and `linkedMetadata`.

`conceptDescription` is a data type property of concept, which refers to the predefined contexts that define and describe an ontological concept. It normally consists of several descriptive phases, which can be used for computing semantic similarity values with metadata (will be talked in the next section).

`associatedMetadata` is an object property of concept, which is used to store the URIs of semantic similar metadata to the concept.

The OWL code of ontological concept format is as below:

```
<owl:Class rdf:ID="Concept" />
  <owl:DatatypeProperty rdf:ID="conceptDescription">
    <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string" /
>
    <rdfs:domain rdf:resource="#Concept" />
  </owl:DatatypeProperty>
  <owl:ObjectProperty rdf:ID="associatedMetadata">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="associatedConcepts" />
    </owl:inverseOf>
    <rdfs:domain rdf:resource="#Concept" />
    <rdfs:range rdf:resource="#Metadata" />
  </owl:ObjectProperty>
```

3.2 Metadata Format

Each metadata has two primary properties (also can be extended), which are metadataDescription and associatedConcepts.

metadataDescription is a data type property of metadata, which refers to the description of a metadata. The content of this property is formed by the Metadata Abstractor, by extracting meaningful information from webpages. Similar to the counterpart in concepts, this property is also used to compute similarity values between metadata and concepts.

associatedConcepts is an object property of metadata, which is used to store the URIs of associated concepts. This property is the inverse of the associatedMetadata property in concepts. In other words, if a metadata stores a concept's URI in the associatedConcepts property, the concept must automatically have the metadata's URI in its associatedMetadata property.

The OWL code of metadata format is as below:

```
<owl:Class rdf:ID="Metadata" />
  <owl:DatatypeProperty rdf:ID="metadataDescription">
    <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string" /
>
    <rdfs:domain rdf:resource="#Metadata" />
  </owl:DatatypeProperty>
  <owl:ObjectProperty rdf:about="#associatedConcepts">
    <owl:inverseOf rdf:resource="#associatedMetadata" />
    <rdfs:domain rdf:resource="#Metadata" />
    <rdfs:range rdf:resource="#Concept" />
  </owl:ObjectProperty>
```

4 Extended Case-Based Reasoning Algorithm

In this section, we will introduce the case-based reasoning algorithm and its extended version adopted in the semantic crawler.

4.1 Case-Based Reasoning (CBR) Algorithm

CBR model is used to retrieve and reuse the existing problem solutions for emerging problems, which has four sub-processes as below [1]:

Retrieve: a new problem is matched with cases in database.

Reuse: if there are matched cases, the solutions to the retrieved cases are reused as the solutions of the emerging problem.

Revise: if the retrieved cases cannot completely match the problems, the solutions to the problem need to be revised.

Retain: the new case, incorporating with both problems and solutions, is stored in database.

Every feature extracted from incident reports is awarded an equal weight. Every feature in a new incident is compared with the corresponding feature in each of the other incidents. If the features match, a score of 1 is awarded. If the features do not match, a score of 0 is awarded. A similarity score is calculated by:

1. Finding the sum of the matching features;
2. Dividing this sum by the number of features contained in the incident, as in the equation (1) below:

$$sim(T, S) = \frac{\sum_{i=1}^n f_i(T, S_i)}{n} \quad (1)$$

Then a threshold is set up to determine whether the two incidents are matched or not.

4.2 Extended CBR Algorithm

Based on the principle of CBR algorithm, we design an Extended CBR algorithm, in order to apply it in the field of information retrieval. The Extended CBR algorithm is used to compute the similarity values between metadata and concepts, by mutually matching the contents of their metadataDescription and conceptDescription properties. If the similarity values are above a predefined threshold, the metadata and concept are determined to be associated.

The pseudocode of Extended CBR algorithm is shown as below:

Input: $M = (k_1, k_2 \dots k_m)$, where M is the description of a metadata consisting of a group of keywords k ; $C = (c_1, c_2 \dots c_n)$, where C is an array of concepts in KB. $c = (d_1, d_2 \dots d_k)$, where d is a concept description of concept c .

Output: C' , where C' is an array of selected concepts based on their semantic similarity values with the metadata M .

Algorithm:

```

BEGIN
  SET C` TO NULL;
  FOR i = 1 TO n
    SET sim TO 0
    FOR j = 1 TO k
      SET aj TO 0
      FOR h = 1 TO m
        IF kh in dj THEN
          ADD 1 TO aj
        END IF
      END FOR
      SET aj TO aj / length of dj
    ENE FOR
    CHOOSE the maximum aj
    SET sim TO aj
    IF sim >= threshold THEN
      PUT ci INTO C`
    END IF
  END FOR
END

```

The Extended CBR model is very simple to implement, and it does not need to generate index terms before matching, which saves the preprocessing time. It also can adapt to the flexibility of concepts that often needs regenerating index terms in most of index term-based algorithms. Since the Algorithm is independent of index terms, it does not have the issue of index term independency.

5 System Implementation and Evaluation

In this section, we will implement the semantic crawling system, and make evaluations for its performance.

5.1 System Implementation

According to the system architecture, the implementation can be divided into two parts – knowledge base and semantic crawler. The knowledge base is built in Protégé-OWL; the semantic crawler is realized in JAVA.

5.2 Crawler Benchmarks

To evaluate the performance of our semantic crawler, we choose a benchmark – precision, and then perform a series of experiments based on it.

Precision for a single concept is the proportion of associated, at the same time, and semantically similar metadata in all associated metadata to the concept, which can be represented as below:

$$\text{Precision}(S) = \frac{\text{number of associated and semantically similar metadata}}{\text{number of associated metadata}}$$

With regard to the whole collection of concepts, the whole precision is the sum of precision for each concept normalized by the number of concepts in the collection, which can be represented in equation (2) below:

$$\text{Precision}(W) = \frac{\sum_{i=1}^n \text{Precision}(S_i)}{n} \tag{2}$$

The purpose of precision is to test the effectiveness of the semantic crawler.

5.3 Experiment

As mentioned before, after the similarity values between a concept and a metadata is obtained, a predefined threshold is used to determine whether the concept and metadata should be associated or not. To obtain the most proper threshold value, our evaluation concentrates on testing the benchmark along with different threshold values.

To test the crawler, we create a transport service ontology with 262 ontological concepts. Then we use our semantic crawler to crawl 400 webpages under the category of transport in the Australian Yellowpages® website. From the 400 webpages, our crawler abstracts 736 metadata in total. The benchmark results are shown in Fig. 2.

From the precision @ threshold figure (Fig. 2), it is observed that the rise of precision is analogous to the rise of threshold – the precision quickly jumps from 15.86% to 93.53% when the threshold rises from 0.5 to 1 (totally 11 values). The precision

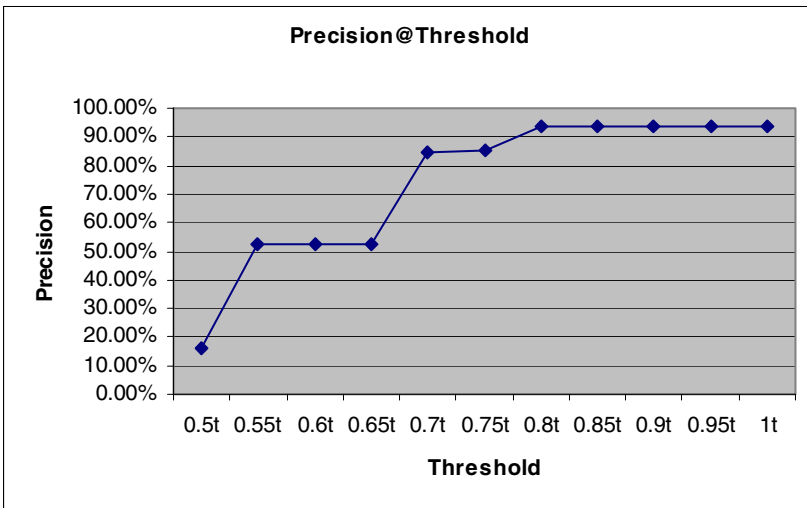


Fig. 2. Precision @ threshold

reaches the top point and keeps steady after the threshold value reaches to 0.8. It is recommended that the threshold must be kept at 0.7 at least, since the precision keeps in a good performance ($\geq 80\%$) in this condition.

By means of observing the benchmark's result, it is concluded that 0.8 could be the most proper threshold value for the semantic crawler, since the precision level jumps 77.67%, compared with its value when the threshold is 0.5. The precision is above 90% in this point, which is a convincing performance in the present stage.

6 Related Works

Generally the semantic crawlers can be divided into two categories. The first category of crawlers is the ones that directly harvest semantic web documents from the internet. Another category is the ones that abstract semantic web documents from the normal web documents – “metadata abstraction crawler” [5].

SWoogle Crawler belongs to the first category, which can harvest, parse and analyze semantic web documents or semantic web information pieces embedded in web documents [3]. A SWoogle Analyzer is used to analyze the content of harvested semantic web documents for indexing and ranking purpose.

Slug is another example of the first category, which is a project for harvesting RDF documents from the internet [4]. It is realized in JENA API, and no performance evaluation details are provided by this project team.

For the metadata abstraction crawlers, the most typical example is Ontobroker. Ontobroker is a crawling system designed with the purpose of extracting, reasoning and generating RDF-annotated metadata [2]. Here an Ontocrawler is used to extract the formal knowledge from HTML web pages. Two different approaches are implemented here. For similarly structured HTML files, a wrapper is used to generate their formal descriptions, by means of referring to an ontology in a knowledge base; for the specially structured HTML files, an annotation language is used.

Handschuh and Staab design a framework of metadata creator – CREAM. A RDF crawler is utilized to find references for created metadata, with the purpose of avoiding duplication [6]. In the CREAM, when the metadata creator wants to find whether an instance already exists or not, the RDF crawler retrieves the instance from the local knowledge base, which stores the RDF files harvested from the semantic web [7]. If a URI with regards to the instance is returned by the RDF crawler, the creator will then be aware that the relational metadata is created [8].

Overall speaking, the semantic crawler's research is still in the beginning stages, and currently not too many semantic crawlers are found. Most semantic crawlers do not provide their evaluation details [5]. However, along with the increasing popularity and maturity of semantic web technologies, it is reasonably believed that the semantic crawler research will soon step into a new era.

7 Conclusions and Further Works

In this paper, we present a semantic crawler based on the extension of the CBR algorithm into the field of semantic information retrieval. The whole crawling system

mainly contains two main parts – a semantic crawler and a knowledge base. The semantic crawler consists of four agents – a Webpage Fetcher, a Webpage Parser, a Metadata Abstractor and a Metadata Matcher. The Webpage Fetcher is responsible for downloading webpages from the internet; the Webpage Parser is adopted for parsing web documents and filtering meaningless information from the downloaded webpages; the Metadata Abstractor is to abstract meaningful information and to use them to build metadata; the Metadata Matcher uses the Extended CBR algorithm to associate the metadata with ontological concepts. The knowledge base is utilized to store the collection of predefined ontological concepts and abstracted metadata. Based on the semantic crawler's mechanism, we design the OWL format for the ontological concept and metadata, with the purpose of realizing metadata abstraction, and metadata-concept association. The CBR algorithm originates from the datamining field, which is used to retrieve and reuse the existing problem solutions for emerging problems. Here we regard concepts as existing problem solutions, and metadata as emerging problems, to apply the algorithm. By comparing the property of conceptDescription from concepts and the property of metadataDescription from metadata, the semantic similarity values between the metadata and concepts can be computed. If the semantic similarity value is greater than a predefined threshold, the metadata and concept can be determined as associated, and the URI of each side is stored into each other's opposite property. Then we realize the semantic crawler by means of Protégé-OWL and JAVA. To test the performance of the semantic crawler, we create a transport service ontology, and choose the Australian Yellowpages® website as testing data source. We use the semantic crawler to crawl 736 metadata from the 400 business webpages under the category of transport in this website. Based on a benchmark – precision from the traditional information retrieval evaluation methods, we find the most proper threshold for the semantic crawler. The figures show that, in the point of this threshold, the precision is above 90%, which is a superior performance in the present stage.

One issue of our research is that the evaluation's size is limited. Therefore our further works will focus on exploring more websites and crawling more webpages for evaluating purpose. We also need to create ontologies in other domains. In addition, more benchmarks will be selected from the information retrieval field, to prove the semantic crawler in multiple perspectives.

References

1. Carthy, D.C.J., Drummond, A., Dunnion, J., Sheppard, J.: The use of data mining in the design and implementation of an incident report retrieval system. In: *Systems and Information Engineering Design Symposium*, pp. 13–18. IEEE, Charlottesville (2003)
2. Decker, S., Erdmann, M., Fensel, D., Studer, R.: Ontobroker: Ontology based access to distributed and semi-structured Information. In: Meersman, R. (ed.) *Database Semantics: Semantic Issues in Multimedia Systems*, pp. 351–369. Kluwer Academic Publisher, Dordrecht (1999)
3. Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V.C., Sachs, J.: Swoogle: a search and metadata engine for the semantic web. In: *The Thirteenth ACM Conference on Information and Knowledge Management*. ACM Press, Washington D.C. (2004)
4. Dodds, L.: *Slug: a semantic web crawler* (2006)

5. Dong, H., Hussain, F.K., Chang, E.: State of the art in metadata abstraction crawlers. In: 2008 IEEE International Conference on Industrial Technology (IEEE ICIT 2008). IEEE, Chengdu (2008)
6. Handschuh, S., Staab, S.: Authoring and annotation of web pages in CREAM. In: WWW 2002, pp. 462–473. ACM Press, Honolulu (2002)
7. Handschuh, S., Staab, S.: CREAM: CREAting Metadata for the Semantic Web. *Computer Networks* 42, 579–598 (2003)
8. Handschuh, S., Staab, S., Maedche, A.: CREAM — Creating relational metadata with a component-based, ontology-driven annotation framework. In: K-CAP 2001, pp. 76–83. ACM Press, Victoria (2001)

Author Index

- Abdelaziz, T. 108
Aldana-Montes, José F. 957, 976
Archimède, Bernard 313
Archner, O. 64
Aréchiga, M.A. 31
Armendáriz-Iñigo, J.E. 914, 924
Arrassi, Allal Zakaria 183
Astudillo, Hernán 324
Ayadi, Nadia Yacoubi 998
- Baatarjav, Enkh-Amgalan 211
Balsters, Herman 699
Béchet, Nicolas 625
Beigl, M. 830
Belk, Mario 595
Berchtold, M. 830
Bernus, Peter 304
Besana, Paolo 965
Bevilacqua, Luca 528
Bhiri, Sami 27, 160
Billhardt, Holger 138
Bindelli, Silvia 76
Bitterberg, Tilmann 118
Bollen, Peter 678, 718
Borg, Marlena 728
Bosque, José Luis 88
Branki, Cherif 108, 118
Braun, Simone 38
Bravo, Crescencio 354, 442
Bravo, Maricela 128, 656
Brochhausen, Mathias 1046
Bürger, Tobias 584
Burgers, Bas 861
- Campbell, Roy H. 883
Capuano, Luisa 528
Carver, Andy 770
Casallas, Rubby 22
Caschera, Maria Chiara 480
Castro Cárdenas, Nicté-Há 241
Centeno, Roberto 138
Cervolo, Paolo 1010, 1066
Cerri, Davide 986
Cerri, Stefano A. 98
Chabot, Joséé 728
- Chang, Elizabeth 1036, 1076
Chang, Hyunjun 851
Charbonnaud, Philippe 313
Chartrand, Éric 728
Chen, David 273
Chen, Jinjun 29
Chniber, Othmane 976
Christiaens, Stijn 230, 615
Chung, Changshin 851
Chung, Lawrence 452
Cichoń, Jacek 904
Cocos, Cristian 1046
Concha, David 283
Conrad, Stefan 40
Contreras-Castillo, Juan 841
Cooper, Kendra M.L. 384, 452
Corchuelo, Rafael 463
Corvino, Fabio 528
Criscione, Claudio 76
Curino, Carlo A. 76
Cysneiros, Luiz Marcio 324
- D'Ulizia, Arianna 509
Damián-Reyes, Pedro 841
Damiani, Ernesto 1010, 1066
Daneva, Maya 241
Dantu, Ram 200, 211, 489, 571
Das, Sajal K. 820
Datta, Anwitaman 16
Davoust, Alan 937
De Francisco Marcos, David 986
De Furio, Ivano 528
de Laaf, Johanna 615
De Leenheer, Pieter 797
de Mello, Carlos E.R. 18
de Mendivil, J.R. González 914, 924
de Uvarow, Simon 1018
Debruyne, Christophe 183, 797
Decker, C. 830
Decker, H. 924
Della Valle, Emanuele 986
Deridder, Dirk 22
Díaz, Alicia 220
Dong, Hai 1076

- Dörr, Martin 1046
 Drago, Mauro L. 76
 Dugenie, Pascal 98
 Duque, Rafael 354
 Durling, Scott 519
 Dzolikhifli, Zarina 548

 El-Bèze, Marc 625
 Elammari, M. 108
 Eliassen, Frank 415
 Ernst, Denise 750
 Esfandiari, Babak 937
 Espadas, Javier 283
 Espinosa, M.E.C. 31
 Everest, Gordon C. 807
 Evripidou, Paraskevas 873
 Eynard, Davide 76

 Faerber, M. 10
 Falkowski, Maciej 36
 Favela, Jesús 374, 841
 Feng, Kunwu 384
 Fernández, Alberto 138
 Fernández, Pablo 463
 Ferreira de Souza, Moisés 18
 Ferri, Fernando 480, 509
 Flores, Roberto A. 499
 Fuentes, Lidia 334

 Gaaloul, Walid 27
 Gal, Avigdor 8, 65
 Gallardo, Jesús 442
 Gattass, Marcelo 12
 Geeverghese, Rajiv 20
 Germanakos, Panagiotis 595
 Gianini, Gabriele 1066
 Giunchiglia, Fausto 986
 Glasspool, David 965
 González, Oscar 22
 Goodwin, Daniel 499
 Gorawski, Marcin 34
 Gottschalk, Florian 263
 Götz, M. 10
 Graf, Norbert 1046
 Grifoni, Patrizia 480, 509
 Guédria, Wided 273

 Hadzic, Maja 1036
 Halpin, Terry 688, 699
 Hauhs, M. 64

 Hauswirth, Manfred 27, 160
 He, Keqing 668
 Hermoso, Ramón 138
 Herold, Sebastian 473
 Herrero, Pilar 88
 Hildmann, Hanno 118
 Hoelz, Bruno W.P. 20
 Horst, Arie 861
 Hurtado, María Visitación 354
 Hussain, Farookh Khadeer 1076

 Ibrahim, Hamidah 548
 Iglér, M. 10
 Ishak, Karim 313

 Jablonski, S. 10
 Jansen-Vullers, Monique H. 263
 Jasiński, Andrzej 904
 Jedrzejek, Czeslaw 36
 Jochaud, F. 10
 Jonquet, Clement 98
 Juiz, Carlos 407
 Junior, Hugo C. 20

 Kamoun, Farouk 407
 Kapelko, Rafał 904
 Kerzazi, Amine 976
 Kessler, Rémy 625
 Khouja, Mehdi 407
 Kim, Dongsun 432
 Kim, Kangtae 364
 Kitagawa, Hiroyuki 6
 Kobayashi, Takashi 14
 Kondratova, Irina 519
 Konstantinou, Ioannis 3
 Kotis, Konstantinos 193
 Koziris, Nectarios 3
 Krummenacher, Reto 986

 Lacroix, Zoé 998
 Lamolle, Myriam 66
 Le-Trung, Quan 415
 Leida, Marcello 1010
 Lekkas, Zacharias 595
 Lemmens, Inge 760
 Li, Canqiang 294
 Li, Qing 294
 Li, Xiang 1
 Liu, Yingbo 24
 Llambías, Guzmán 1018
 López, Claudia 324

- Lorenzetti, Carlos M. 646
 Lumsden, Joanna 519, 538

 Ma, Weimin 452
 MacLean, Ryan 538
 Madiraju, Praveen 548
 Magaldi, Massimo 528
 Maguitman, Ana G. 646
 Martín, Luis 1046
 Mata, Susana 88
 Mawlood-Yunis, Abdul-Rahman 894
 Mazón, Jose-Norberto 44
 McQuinn, Jerre 738
 Meersman, Robert 1056
 Melli, Gabor 738
 Meratnia, Nirvana 861
 Milovanović, Miloš 561
 Minović, Miroslav 561
 Molina, Arturo 283
 Moreira de Souza, Jano 18
 Motz, Regina 1018
 Mourlas, Constantinos 595
 Muñoz, Lilia 44
 Muñoz-Escoí, F.D. 914, 924

 Nagypal, Gabor 38
 Naor, Dalit 986
 Naudet, Yannick 273
 Navas-Delgado, Ismael 976
 Nijssen, Maurice 760
 Nixon, Lyndon 947, 986
 Noguera, Manuel 354
 Noran, Ovidiu 304
 Normann, Ragnar 780

 Obermeier, Philipp 986
 Ohki, Kosuke 6
 Oliveira, Jonice 18
 Ontañón, Santi 150
 Orsi, Giorgio 76
 Ortiz, Rubén 138
 Ossowski, Sascha 138

 Pardillo, Jesús 44
 Park, Sooyong 432
 Patkar, Vivek 965
 Peeters, Johannes 183
 Penzenstadler, Birgit 426
 Pereira, Carla 605
 Phithakkitnukoon, Santi 200, 211, 571

 Pierno, Stefania 528
 Pinto, Mónica 334
 Piprani, Baba 668, 728, 750
 Plaza, Enric 150
 Pluciennik, Ewa 34
 Press, Martin 499
 Puder, A. 830
 Puigjaner, Ramon 407
 Pulido, J.R.G. 31

 Qiao, Ying 1

 Ralha, Célia G. 20
 Raposo, Alberto B. 12
 Rebholz-Schuhmann, Dietrich 986
 Redondo, Miguel Á. 442
 Ren, Kaijun 29
 Resinas, Manuel 463
 Riedel, T. 830
 Roa-Valverde, Antonio J. 957
 Robertson, Dave 965
 Robertson, Edward L. 253
 Roche, Mathieu 625
 Rodríguez, Marcela D. 374
 Rodríguez, María Luisa 354
 Rohjans, Sebastian 1026
 Romano, Luigi 528
 Romero, David 283
 Rouvoy, Romain 415
 Roy, Nirmalya 820
 Ruiz-Fuertes, M.I. 924
 Russo, Roberto 528

 Salinas, R. 914
 Samaras, George 595
 Sánchez, Christian 635
 Santana Tapia, Roberto 241
 Santos, Ismael H.F. 12
 Schlüter, Tim 40
 Scholz, Thorsten 170
 Schulte, Stefan 1026
 Scotto di Carlo, Vladimiro 528
 Sheremetov, Leonid 635
 Shin, Seokkyoo 851
 Shu, Lei 27
 Simperl, Elena 584, 986
 Skagestein, Gerhard 780
 Smith, Barry 1046
 Soares, António Lucas 605
 Song, Junqiang 29

- Springer, John A. 253
 Spyns, Peter 1056
 Starčević, Dušan 561
 Štavljanin, Velimir 561
 Steinmetz, Ralf 1026
 Stencel, Krzysztof 396
 Stenzhorn, Holger 1046
 Ströele A. Menezes, Victor 18
 Subramanian, Nary 344

 Tadeu da Silva, Ricardo 18
 Taherkordi, Amirhosein 415
 Tang, Yan 384, 787
 Teymourian, Kia 947, 986
 Tian, Kun 384
 Timm, Ingo J. 170
 Toledo, Federico 1018
 Torres-Moreno, Juan Manuel 625
 Trog, Damien 230, 615, 787
 Trujillo, Juan 44
 Tsianos, Nikos 595
 Tsiknakis, Manolis 1046
 Tsoumakos, Dimitrios 3
 Tziakouris, Marios 873

 Uslar, Mathias 1026

 van Brandenburg, Ray 861
 Van Damme, Céline 230
 Van de Maele, Felix 220, 1056
 van der Aalst, Wil M.P. 263
 van Eck, Pascal 241
 van Oene, Leida 241
 Vasirani, Matteo 138

 Velazquez, José 128, 656
 Vidal, Maria-Esther 998
 Volz, Bernhard 54
 Vos, Jos 709
 Vulcu, Gabriela 160

 Walter, Andreas 38
 Wang, Chaokun 24
 Wang, Chong 668
 Wang, Hongan 1
 Wang, Jianmin 24
 Wang, Yun 294
 Watanabe, Tetsutaro 14
 Watanabe, Yousuke 6
 Węgrzynowicz, Patrycja 396
 Weiler, Gabriele 1046
 Widemann, B. Trancón y 64
 Wu, Victor K.Y. 883

 Xiao, Nong 29
 Xin, Liu 16

 Yang, Hedong 24
 Yokota, Haruo 14

 Zaiß, Katrin 40
 Zawada, Marcin 904
 Zerdazi, Amar 66
 Zhang, Huiqi 489
 Zhao, Gang 615
 Zhong, Kang 1
 Zhou, Zhangbing 27, 160
 Zimbrão, Geraldo 18