J Netw Syst Manage (2009) 17:309–325
DOI 10.1007/s10922-009-9132-6

Use of Ontologies as Representation Support of Workflows Oriented to Administrative Management

Álvaro E. Prieto · Adolfo Lozano-Tello

Published online: 27 May 2009

Abstract One of the applications of workflow systems is the management of administrative processes characterized by the transmission of information elements among users of an organization. Tasks contained in these processes are carried out by users responsible for confirming, modifying or adding information throughout. These processes need to be defined in workflow management systems in which all the elements are perfectly identified and are easily adaptable to changes that may arise in the sequences of tasks, in the users involved or in the data transmitted from one task to another. For this kind of processes is easier to reuse those represented in ontologies. On one hand, existing ontologies for representing some domain elements can be reused. At the same time, ontologies have an excellent expressive capacity to define tasks, their relationships and the flow control among them with precision. This paper proposes a complete model, together with the necessary software tools, for tackling this issue.

Keywords Workflows · Ontologies · Business process · Administrative management processes

1 Introduction

"Workflow is the automation of a business process, in whole or in part, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules" [1]. A Workflow Management

Á. E. Prieto · A. Lozano-Tello
Computing Systems and Telematics Engineering Department,
University of Extremadura, Cáceres, Spain

Á. E. Prieto (✉)
Escuela Politécnica de Cáceres, Avda de la Universidad sn, 10071 Cáceres, Spain
e-mail: aeprieto@unex.es

System (WfMS) is able to interpret the workflow definition and creates and manages the execution of these workflows. Specifically, van der Aalst [2] states that “a WfMS ensures that the right information reaches the right person at the right time, or is submitted to the right computer application at the right moment’’. Workflow Management Coalition (WfMC[1]) is making a great effort to standardize these systems to facilitate their widespread application. Moreover, in the last few years, WfMS are gaining popularity thanks to reports that identify Workflow Management Coalition (BPM) as the number one business priority [3].

Traditionally, workflows are classified in three kinds of workflow [4]: ad hoc, administrative, and production. Ad hoc workflows perform office processes, such as product documentation or sales proposals, where there is no set pattern for moving information among people. Administrative workflows involve repetitive, predictable processes with simple task coordination rules, such as routing an expense report or travel request through an authorization process. The ordering and coordination of tasks in administrative workflows can be automated. Production workflows involve repetitive and predictable business processes, such as loan applications or insurance claims. Unlike administrative workflow, production workflows typically encompass a complex information process involving access to multiple information systems.

On the other hand, Georgakopoulos et al. [5] characterize workflow along a continuum from human-oriented to system-oriented. At one extreme, human-oriented workflow involves humans collaborating in performing tasks and coordinating tasks. The requirements for WfMSs in this environment are to support the coordination and collaboration of humans and to improve human throughput. Humans, however, must ensure the consistency of documents and workflow results. This kind of workflows requires particular graphical user interface concepts. The main concept is the work item list that is used by workers in order to interact with the system. At the other extreme, system-oriented workflow involves computer systems that perform computation-intensive operations and specialized software tasks.

Lastly, Weske [6] classifies workflows in single-application or multiple-application. A single application workflow consists of activities and their causal and temporal ordering that are realized by one common application system. A multiple-application workflow contains activities that are carried out by multiple application systems, providing an integration of these systems.

Certain types of simple business processes exist, generally in the administrative or legal ambit, characterized by the sequential transmission among users of straightforward and well structured information elements in which the user or users in charge of the task must confirm, modify or add new information until said process ends. This type of processes does not usually require a complex WfMS with advanced characteristics such as coordination with other external applications or different possibilities in the process flow control or in its events. Basically this type of processes needs to be supported by workflow management systems, which in turn, need to be easily adaptable to changes that may occur either in the task sequences, the users involved or in the data transmitted among users. As an example

[1] http://www.wfmc.org.

of this type of processes, we may cite all those initiated by a user and which must be attended to or evaluated by other different users following a perfectly defined protocol for data, times and agents involved. This would refer to processes laid down by laws, rulings of action or protocol executions in public institutions or large companies. Examples could be the management of public contest bids, holiday application procedures or notification of an incident in a company's information system.

Concerning the previous classifications, the workflow system that manages this type of process is an administrative, human-oriented and single-application workflow system and which additionally is easily adaptable to the changes in the data elements that it is managing.

With the aim of managing processes with these characteristics, this paper proposes a model that represents the definition of business processes by means of ontologies, and at the same time, presents a WfMS which works with such representations. This proposal enables the perfect identification and easy adaptation of the data used, the processes which manage it and the users involved in said processes thanks to the advantages of integration and reutilization that the ontologies provide.

This paper is structured as follows: Sect. 2 presents existing work that apply ontologies to workflow management systems and Sect. 3 details the model composed of ontologies and the WfMS.

2 Work on Ontologies Applied to Workflows

In recent years, some interesting approaches have appeared in ontological engineering applied to improving WfMS. The most recognized are:

Vieira et al. [7] is one of the first works integrating both fields in 2004. This paper proposes a solution to make workflow execution more flexible in the presence of incomplete information, by adopting presuppositions, and in the presence of negative information, by suggesting execution alternatives. This paper also presents "an architecture for the workflow system, which is driven by ontologies that capture semantic relationships between workflows, resources and users".

In that year, Pathak et al. [8] develop ontology-extended workflow components and mappings between ontologies to facilitate assembly of ontology-extended, component-based workflows using semantically heterogeneous workflow components. The proposed ontology-extended component-based workflows provide a theoretical framework for the assembly of semantically well-formed workflows from semantically heterogeneous information sources and software components.

One year later, Zdravkovic and Kabilan [9] propose a methodology which, using a layered contract ontology, deduces contract requirements into a high-level process description named Contract Workflow Model (CWM). By applying a set of transformation rules, the CWM is then compared for compliance with existing, executable process models. By the use of its concepts, the methodology enables comprehensive identification and evolution of requirements for interoperability of processes of the contracting parties.

In 2006, one work that stands out is the work of Gasevic and Devedzic [10]. The main idea of this paper is that the Petri net ontology should provide the necessary Petri net infrastructure for the Semantic Web. The infrastructure understands Petri nets sharing using XML-based ontology languages (i.e., RDFS and OWL). Petri nets have been widely applied in systems, modeling and analysis for many years.

Haller et al. [11] present a multi meta-model process ontology (m3po), which is based on various existing reference models and languages from the workflow and choreography domain. The mp3o ontology relates workflow models to choreography models and allows choreography extraction from internal workflow models.

Vidal et al. [12] outline a framework that extends the Unified Problem-solving Method description Language in order to enhance workflow modeling with knowledge. The framework defines the knowledge components needed to represent and reuse both the static and dynamic knowledge used to describe a business process.

In 2007, Yao et al. [13] introduce a workflow centric collaboration system based on ontologies, which is context-aware and adaptive. Using ontologies to represent most collaboration elements and rules of the system, they introduce an ontology repository into the framework.

Abramowicz et al. [14] present a semantically enhanced Business Process Modeling Notation [15], namely the sBPMN ontology. The sBPMN ontology overcomes problems with composition and execution of processes based on the models designed by business analysts. sBPMN proposes to use ontologies as a fundamental basis and it provides not only hierarchy, but also axioms and mapping to an upper process ontology (UPO). And also in that year, Andonoff et al. [16] propose a coordination protocol ontology for Inter-Organizational Workflow and explains how workflow partners can select them dynamically.

In general, each one of these approaches uses ontologies in the workflow field incorporating the benefits of reuse, consistency and shared consensus knowledge of ontologies. Our approach applies the advantages of ontologies to the administrative management processes domain where managers may define the processes, existing processes are easily reused, and the case data and users may be modified and incorporated into the system without modifying the definition of processes.

3 A Workflow Model Based on Ontologies for Processes of Administrative Management

3.1 Ontologies for the Representation of Workflows

In recent years, when it is necessary to represent knowledge in any domain, the use of ontologies is becoming ever more frequent, both in reasoning and knowledge-based systems as well as in traditional information systems. Ontologies have been used in the field of Artificial Intelligence for Knowledge Engineering, in natural language processing and in basic knowledge representation. Today they are also being widely used in fields such as intelligent information integration, cooperative information systems development, information retrieval, e-commerce and knowledge management [17].

The reason for the success of ontologies is derived from their design criteria, which was described by Gruber [18]: clarity, coherence, extendibility, minimal encoding bias and minimal ontological commitment. Following this approach, ontologies are designed with the aim of their knowledge being easily reusable and shared by the communities of the same domain. In this way, the taxonomy of concepts which represents a domain should be complete, with properties and relationships that categorize all terms precisely, and with definitions in natural language to describe their meaning. Therefore, ontologies can be used by software agents to exploit the knowledge represented and also by people in order to share the vocabulary in a domain. Recently, the general acceptance of a single language of representation such as OWL[2] and the development of tools like Protégé[3] for the construction of ontologies has favored the wide use of ontologies in many fields, especially in the Semantic Web.

Among other uses, ontologies are used as a common element of unification and integration of information. For example, ontologies are used to integrate heterogeneous relational databases [19] or as a basis to characterize process integration within enterprises [20]. In these cases, ontologies provide a complete, precise and shared terminology about a particular domain which facilitates integration and which will be easily reusable by the same or another organization. These advantages provide a considerable saving of time and effort in processes and data definition tasks, or in merging methods when similar representations of the same domain exist.

Using the previous ideas, our model proposes the application of ontologies for the representation of management processes defined in workflows and case data which are managed by workflow tasks. Although several consolidated models and languages of workflow representation exist [21–24], the application of ontologies in this field, used directly or as a definition of a metalanguage, can provide the following advantages:

- Case data managed by tasks can be exchanged without the need to carry out any modification in the definition of tasks. The taxonomies represented in ontologies, together with the attributes and relationships, are modular elements which can be modified if the domain specifications changes. This implies that, if a task manages some case data represented in an ontology, and if the ontology changes, for example, adding a new subclass, modifying a particular attribute or the values of the attributes, the workflow definition need not be modified. Furthermore, as long as the superclass which links with the task is respected, and in order to deal with a similar workflow but with different case data, a new hierarchy of concepts can be added without the user having to carry out any additional process.
- Definitions of workflow processes, represented in ontologies, are easily reusable. By following ontology development methodologies, users can obtain complete,

[2] http://www.w3.org/2004/OWL/.

[3] http://protege.stanford.edu/.

precise and shared process definitions. This implies that workflow specifications would be more reusable and also, if workflow integration processes of similar domains are necessary, these processes would be less costly.

As mentioned above, ontologies are built to be easily reused, and also, they are usually released on the web without charge. The reuse process may involve some effort, mainly in the processes of search, selection, and in some cases, adaptation to the new system. These factors are discussed in detail in [25].

3.2 Model Overview

The model presented in this paper proposes the use of ontologies to define and manage processes. The problems it deals with refer to management processes where different kinds of users add information in each one of the defined tasks in the workflow. They are management processes that do not require the execution of external applications nor modules that carry out computational operations. Basically, the model proposes how the supervisor user must define the workflow with the tasks that a management process comprises, what information is needed to go from one task to another, and which kind of users are responsible of add these information.

As already mentioned, the model proposes the use of ontologies to define process tasks, the taxonomy of case data transmitted form one task to another, and the taxonomy of users which may add new information. This ensures that the processes are well defined and are more reusable and, in addition, the classes of case data involved and the classes of users related with the tasks can be modified without changing the representation of the workflow process. Moreover, it should be noted that the taxonomies of classes and instances that may be needed as case data, can be defined in ontologies in the organization itself or can be reused from ontology repositories. If we use a hardware incident management workflow as an example, which needs, among other case data, “types of hardware and software components”, the workflow designer could reuse an existing ontology on that domain, with a consequent saving of time.

The proposed solution presents a series of components that form the architecture of the model (Fig. 1). As described in each one of the following subsections, our approach provides an ontology as the basis of workflow representation, together with methods (and their respective software tools) to identify and to exploit the workflows of a management process. The elements that constitute the model are:

1. an ontology of workflows (OntoWorkflow), based on the recommendations of WfMC, which models the elements that compose the administrative management processes and their relationships. OntoWorkflow serves as a framework for defining the terms involved in the business process and the processes that compose it.
2. a method to obtain the ontology of terms (OntoTerms) that consists of the elements to be used as case data and users of the business process which will be managed and built following the specifications of OntoWorkflow.

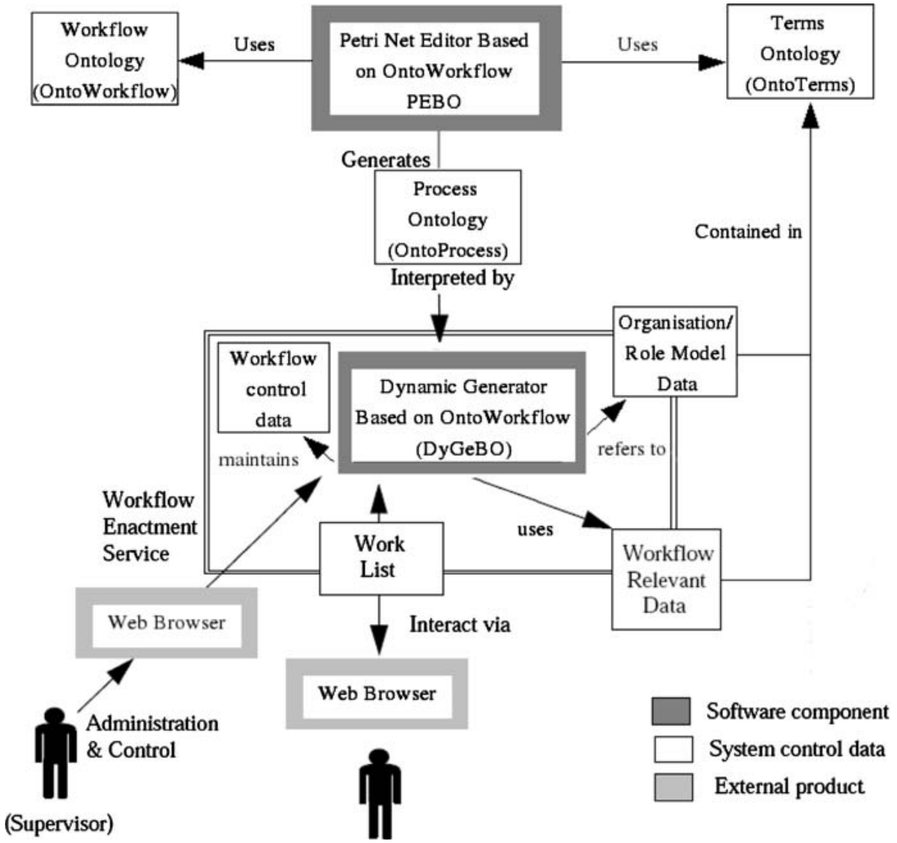


Fig. 1 Overview of the proposed model

3. a method to build the ontology of the process (OntoProcess) that will contain the different states of the process together with tasks that will be necessary to carry it out, the task execution order and the relationships with the different case data and workflow participants previously defined in OntoTerms. An implementation of this method, called PEBO (PetriNet Editor Based on OntoWorkflow), is also provided. PEBO facilitates the representation of administrative processes in an ontology of process using OntoWorkflow, OntoTerms and Petri net.
4. a web application called DyGeBO (Dynamic Generator of WfMS based on OntoWorkflow) which manages the administrative process represented in the ontologies built using OntoWorkflow, i.e., OntoTerms and OntoProcess. The main feature of DyGeBO is that if at any time there is a change in the business process, either in the data elements or in the number or order of the tasks to be performed, DyGeBO is able to rebuild the entire WfMS without affecting the user. That is, it is possible to change OntoProcess, which defines the process, or OntoTerms, which contains the information shown or needed to be modified in every task, and said changes are dynamically applied by DyGeBO.

In the following subsections, the different elements of our model are detailed. As an example, one of the standard management processes of ITIL(Information Technology Infrastructure Library[4]) is used, in particular, the Incident Management Process in Information Systems.

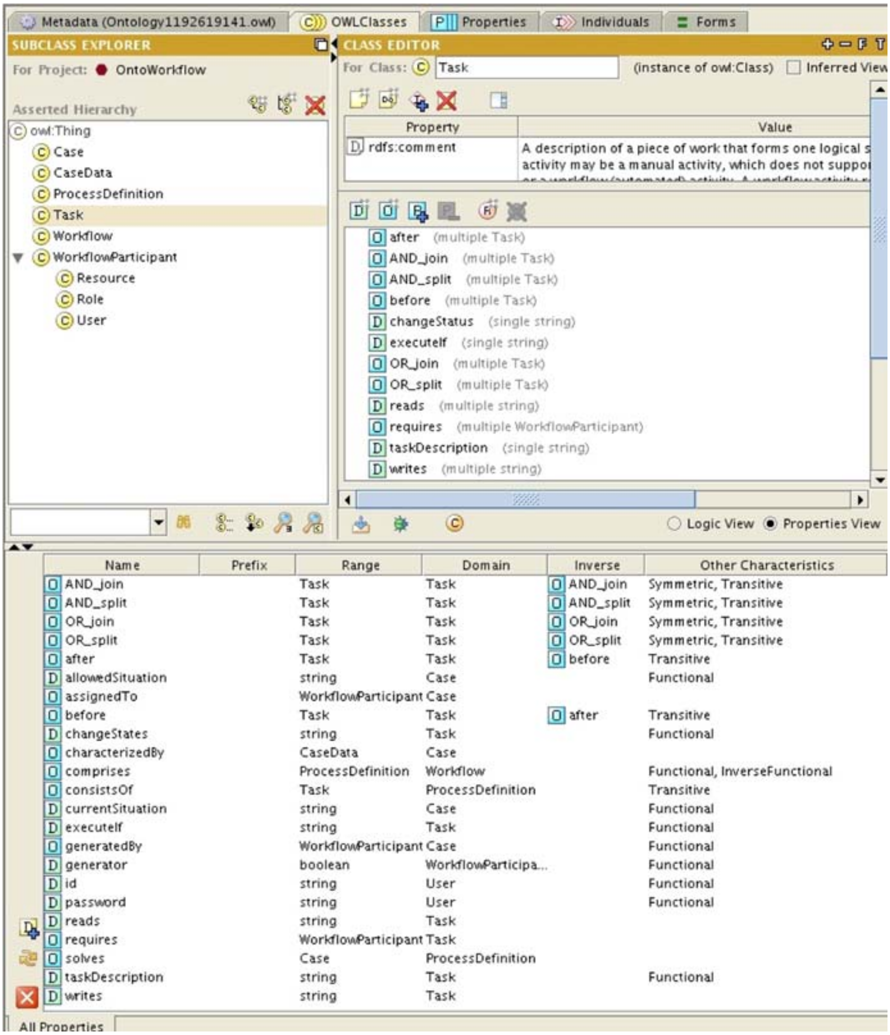
3.3 OntoWorkflow: An Ontology in the Workflow Domain

OntoWorkflow (Fig. 2) is the ontology that contains the elements of a workflow and their relationships. This ontology is built using the definitions of workflow elements provided by the WfMC as knowledge source. It has been developed following METHONTOLOGY methodology [26] and it is represented using OWL Language.[5] OntoWorkflow ontology contains all concepts and relationships needed to represent workflows. The main terms of this ontology are described below:

- Workflow as a description of the problem to be solved and which comprises one or several process definitions.
- Process Definition consists of a number of Tasks that need to be carried out to successfully complete a Case.
- Case represents the services managed by a workflow process. A Case is characterized by Case Data and is assigned to one of the possible participants in the Workflow (Workflow Participant). One of the participants will be the one who will have initiated a new instance of the Case (stored in generatedBy property). Moreover, it has two properties with information about the Case states. The first one (allowedSituation) stores the possible states that a Case has, while the second one (currentSituation) stores the current states of an instance of Case.
- Task is a logical unit of work. Tasks are related among themselves by means of before and after relationships that mark the execution order of tasks. It is also possible to indicate a parallel routing or selective routing in the process execution using the and/or conjunctions in these relationships. A Task may require a Workflow Participant to be performed. The executeIf property maintains the states of the Case that launch the task and the changeStates indicates what the next states of the Case are when the task is finished. Reads and writes properties indicate which Case attributes and Case Data must be shown to the user and which of them must be modified by the user in the task.
- Case Data is the class where the information related with the Case is stored. Each one of these information elements may include not only data values, but also instances of class which in turn contain a set of attributes. An example of this kind of properties in incidents management domain would be the incident resolution priority, with attributes such as price or maximum resolution time. The final workflow system will manage the instances of these classes, therefore, they should be created once these classes have been defined. Thus, following the priorities example, the workflow would have an instance for each one of the priorities supported in incident resolution, together with their respective attributes.

[4] http://www.itil-officialsite.com/.

[5] http://quercusseg.unex.es/ontoquercus/?download=OntoWorkflow.owl.



For Project: ● OntoWorkflow

For Class: Task (instance of owl:Class) □ Inferred View

- owl:Thing
  - Case
  - CaseData
  - ProcessDefinition
  - Task
  - Workflow
  - WorkflowParticipant
    - Resource
    - Role
    - User

Property	Value
rdfs:comment	A description of a piece of work that forms one logical s activity may be a manual activity, which does not suppo

after (multiple Task)
AND_join (multiple Task)
AND_split (multiple Task)
before (multiple Task)
changeStatus (single string)
executeIf (single string)
OR_join (multiple Task)
OR_split (multiple Task)
reads (multiple string)
requires (multiple WorkflowParticipant)
taskDescription (single string)
writes (multiple string)

○ Logic View ● Properties View

Name	Prefix	Range	Domain	Inverse	Other Characteristics
AND_join		Task	Task	AND_join	Symmetric, Transitive
AND_split		Task	Task	AND_split	Symmetric, Transitive
OR_join		Task	Task	OR_join	Symmetric, Transitive
OR_split		Task	Task	OR_split	Symmetric, Transitive
after		Task	Task	before	Transitive
allowedSituation		string	Case		Functional
assignedTo		WorkflowParticipant	Case		
before		Task	Task	after	Transitive
changeStates		string	Task		Functional
characterizedBy		CaseData	Case		
comprises		ProcessDefinition	Workflow		Functional, InverseFunctional
consistsOf		Task	ProcessDefinition		Transitive
currentSituation		string	Case		Functional
executeIf		string	Task		Functional
generatedBy		WorkflowParticipant	Case		Functional
generator		boolean	WorkflowParticipa...		Functional
id		string	User		Functional
password		string	User		Functional
reads		string	Task		
requires		WorkflowParticipant	Task		
solves		Case	ProcessDefinition		
taskDescription		string	Task		Functional
writes		string	Task		

Fig. 2 Snapshot of OntoWorkflow in Protégé

- Workflow Participant is any type of agent needed for performing a task. This class will be composed by the class taxonomy of users which may intervene in each task. Defined in OntoWorkflow, the id and password properties will be used by such users for accessing the workflow system.

3.4 OntoTerms: Ontologies of Workflow Terms

In OntoTerms every element belonging to the domain managed by the workflow is defined. These elements are the case data and the users involved in the workflow. For example, if a business process is about incident management, then a terms ontology is required to describe incident domain elements. Such elements would be computer material which could have the incident, the different priorities that could be assigned in resolution and the different kind of workflow participants which could be involved in the process.

The root elements of this ontology of terms are linked with superclasses defined in OntoWorkflow ontology. That is, each element defined in the domain will be a subclass of some of the elements defined in OntoWorkflow.

The development of this ontology can be carried out with any application that allows ontologies in OWL to be modeled, like Protégé. The simple steps to develop it are:

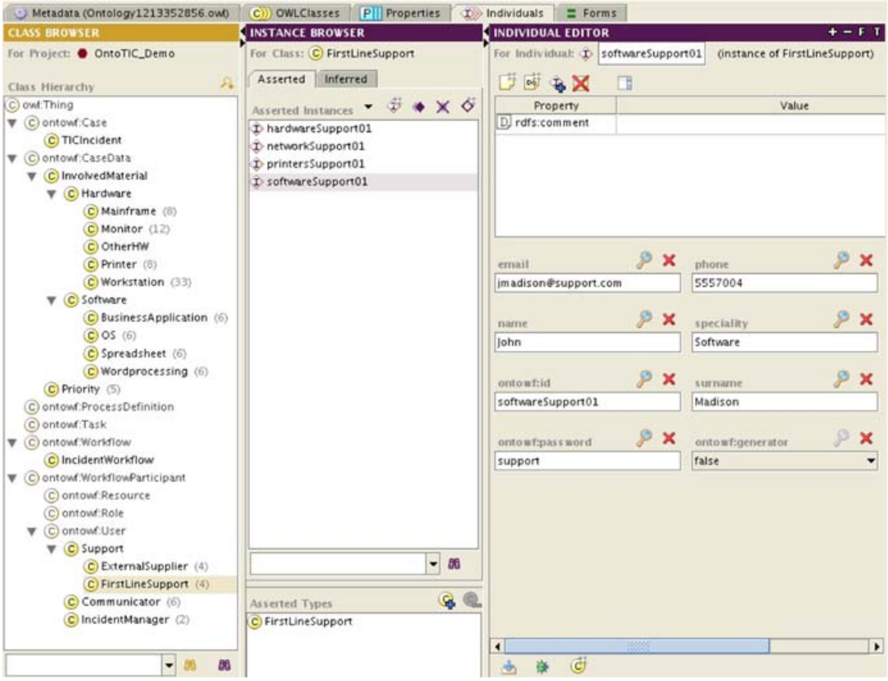
1. Import OntoWorkflow so its elements will be superclasses in OntoTerms.
2. Establish class domain name as subclass of Workflow class.
3. Identify case type to be solved as subclass of Case.
4. Identify the Case attributes as OWL datatype property. An example of these attributes is the incident starting date or its description.
5. Identify Case Data and define their instances. In the example of incident management[6] (Fig. 3), within the Case Data we could have the incident resolution priority with attributes such as price or maximum resolution time. The final workflow system will manage the instances of these classes, therefore, they should be created once these classes have been defined. It is important to point out, that in many domains, well constructed and complete ontologies are uploaded on internet which can be reused, and with subsequent saving in development time. It would be enough to put said ontology as a subclass of Case Data for the WfMS to automatically deal with all the elements defined in it.
6. Identify the different kinds of Workflow Participant and their instances. In the example of incident management there will be incident communicators, supervisors, support users, etc. It is necessary to be careful when defining the instances of these types of users because each instance identifies a potential user of the final WfMS. As mentioned above, for companies which deal with several workflows, the classification of participants represented in an ontology could be simply added or adapted to the new workflow, indicating that the elements are subclasses of WorkflowParticipant.

3.5 OntoProcess: Ontologies of Workflow Processes

Using the specifications of elements and relationships of OntoWorkflow, and the terms and users of the domain of OntoTerms, the processes of the business model are represented in OntoProcess. The order of execution of tasks, the elements of the ontology of terms to be shown, the elements to be added or modified and what participants are to intervene in each task will be represented in this ontology.

The steps of the proposed method to build the ontology of process are detailed below. Logically, these processes should be assisted by a software tool, such as that

[6] http://quercusseg.unex.es/ontoquercus/?download=OntoTIC_Demo.owl.



For Project: ● OntoTIC_Demo

- owl:Thing
  - ontowf:Case
    - TICIncident
  - ontowf:CaseData
    - InvolvedMaterial
      - Hardware
        - Mainframe (8)
        - Monitor (12)
        - OtherHW
        - Printer (8)
        - Workstation (33)
      - Software
        - BusinessApplication (6)
        - OS (6)
        - Spreadsheet (6)
        - Wordprocessing (6)
    - Priority (5)
  - ontowf:ProcessDefinition
  - ontowf:Task
  - ontowf:Workflow
    - IncidentWorkflow
  - ontowf:WorkflowParticipant
    - ontowf:Resource
    - ontowf:Role
    - ontowf:User
      - Support
        - ExternalSupplier (4)
        - FirstLineSupport (4)
      - Communicator (6)
      - IncidentManager (2)

For Class: FirstLineSupport

- hardwareSupport01
- networkSupport01
- printersSupport01
- softwareSupport01

Asserted Types: FirstLineSupport

For Individual: softwareSupport01 (instance of FirstLineSupport)

rdfs:comment

email: jmadison@support.com
phone: 5557004
name: John
speciality: Software
ontowf:id: softwareSupport01
surname: Madison
ontowf:password: support
ontowf:generator: false

Fig. 3 Example of OntoTerms for incident management edited with Protégé

proposed in Subsect. 3.6 with the PEBO tool (PetriNet Editor Based-on OntoWorkflow). These steps are as follows:

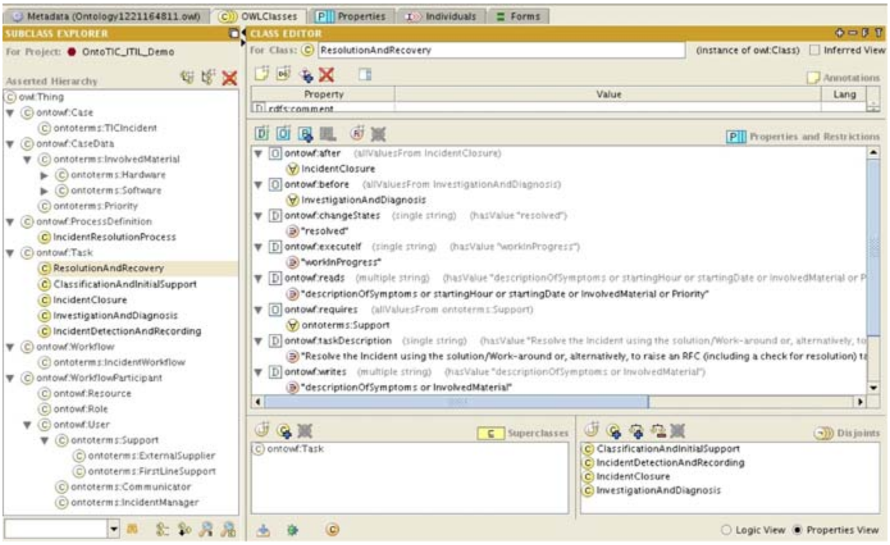
1. Use the ontology of terms of the domain (OntoTerms) built from OntoWorkflow.
2. Identify the process that resolves this type of case as a subclass of class Process indicated in OntoWorkflow.
3. Define the different states of the process using has Value restriction on property AllowedSituation of Case.
4. Define each task of the process as subclass of class Task of OntoWorkflow. For each task it is necessary to:

(a) Describe in natural language the actions taken by the task in the property task Description. This must be done in order to facilitate reuse and consensus about what the task represents.

(b) Indicate which states of those defined in the third step of the method launches the task. This will be represented in the hasValue restriction on executeIf property.

(c) Indicate which states of those defined in the third step of the method has the process when the task finishes. This will be represented in the hasValue restriction on changeStates property.

(d) Indicate which tasks precede it, using AllValuesFrom restriction on *before* relationship. If the task is an or-join or and-join of precedent tasks then



```
For Project: ● OntoTIC_ITIL_Demo
Asserted Hierarchy
owl:Thing
▼ ontowf:Case
    ontoterms:TICIncident
▼ ontowf:CaseData
  ▼ ontoterms:InvolvedMaterial
    ▶ ontoterms:Hardware
    ▶ ontoterms:Software
    ontoterms:Priority
▼ ontowf:ProcessDefinition
    IncidentResolutionProcess
▼ ontowf:Task
    ResolutionAndRecovery
    ClassificationAndInitialSupport
    IncidentClosure
    InvestigationAndDiagnosis
    IncidentDetectionAndRecording
▼ ontowf:Workflow
    ontoterms:IncidentWorkflow
▼ ontowf:WorkflowParticipant
    ontowf:Resource
    ontowf:Role
  ▼ ontowf:User
    ▼ ontoterms:Support
        ontoterms:ExternalSupplier
        ontoterms:FirstLineSupport
      ontoterms:Communicator
      ontoterms:IncidentManager
```

```
For Class: ResolutionAndRecovery          (instance of owl:Class)  □ Inferred View

Property          Value          Lang
rdfs:comment

▼ ontowf:after  (allValuesFrom IncidentClosure)
    IncidentClosure
▼ ontowf:before  (allValuesFrom InvestigationAndDiagnosis)
    InvestigationAndDiagnosis
▼ ontowf:changeStates  (single string)  (hasValue "resolved")
    "resolved"
▼ ontowf:executeIf  (single string)  (hasValue "workInProgress")
    "workInProgress"
▼ ontowf:reads  (multiple string)  (hasValue "descriptionOfSymptoms or startingHour or startingDate or InvolvedMaterial or P
    "descriptionOfSymptoms or startingHour or startingDate or InvolvedMaterial or Priority"
▼ ontowf:requires  (allValuesFrom ontoterms:Support)
    ontoterms:Support
▼ ontowf:taskDescription  (single string)  (hasValue "Resolve the Incident using the solution/Work-around or, alternatively, to
    "Resolve the Incident using the solution/Work-around or, alternatively, to raise an RFC (including a check for resolution) t
▼ ontowf:writes  (multiple string)  (hasValue "descriptionOfSymptoms or InvolvedMaterial")
    "descriptionOfSymptoms or InvolvedMaterial"

Superclasses:
ontowf:Task

Disjoints:
ClassificationAndInitialSupport
IncidentDetectionAndRecording
IncidentClosure
InvestigationAndDiagnosis

○ Logic View  ● Properties View
```

Fig. 4 Example of OntoProcess for incident management edited with Protégé

separate the precedent tasks in the restriction using the conjunction *or* or *and* respectively.

(e) Indicate which tasks follow it, using AllValuesFrom restriction on *after* relationship. If the task is an or-split or and-split of following tasks then separate the following tasks in the restriction using the conjunction *or* or *and* respectively.

(f) Indicate which Case properties and which CaseData (represented in OntoTerms) must be shown to the user in the task. This will be represented in the hasValue restriction on *reads* property.

(g) Indicate which Case properties and which CaseData (represented in OntoTerms) must be added or modified by the user in the task. This will be represented in the hasValue restriction on *writes* property.

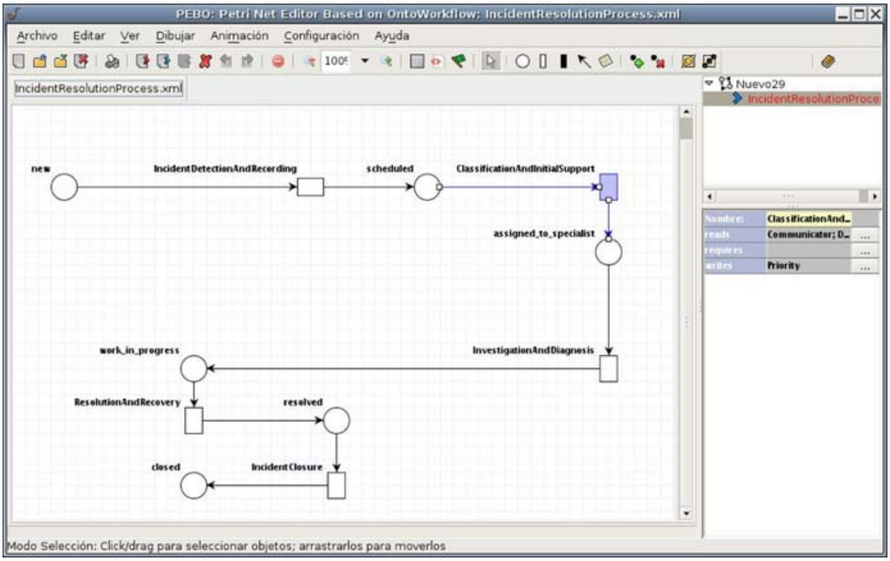
(h) Indicate which kind of participants (represented in OntoTerms) can carry out the task. This will be represented in AllValuesFrom restriction on *requires* relationship.

In Fig. 4, an example of ontology of process for incident management[7] is shown.

3.6 PEBO Tool: PetriNet Editor Based-on OntoWorkflow

As mentioned, the scope of our model is administrative management processes. Those responsible for designing the workflow of their organization need a simple and intuitive tool for performing this process. With this in mind, the process modeling tool called PEBO (PetriNet Editor Based-on OntoWorkflow) has been developed. PEBO allows business processes to be modeled in a Petri net and stores

[7] http://quercusseg.unex.es/ontoquercus/?download=OntoTIC_ITIL_Demo.owl.


PEBO: Petri Net Editor Based on OntoWorkflow: IncidentResolutionProcess.xml
Archivo
Editar
Ver
Dibujar
Animación
Configuración
Ayuda
100%
IncidentResolutionProcess.xml
new
IncidentDetectionAndRecording
scheduled
ClassificationAndInitialSupport
assigned_to_specialist
work_in_progress
InvestigationAndDiagnosis
ResolutionAndRecovery
resolved
closed
IncidentClosure
Nuevo29
IncidentResolutionProce

Nombre:	ClassificationAnd...	
reads	Communicator; D...	...
requires		...
writes	Priority	...

Modo Selección: Click/drag para seleccionar objetos; arrastrarlos para moverlos

Fig. 5 PEBO workplace designing processes for incident management

the elements, using Jena,[8] in the ontology of process i.e., OntoProcess. In Fig. 5, the workplace of this tool with the incident management example is shown.

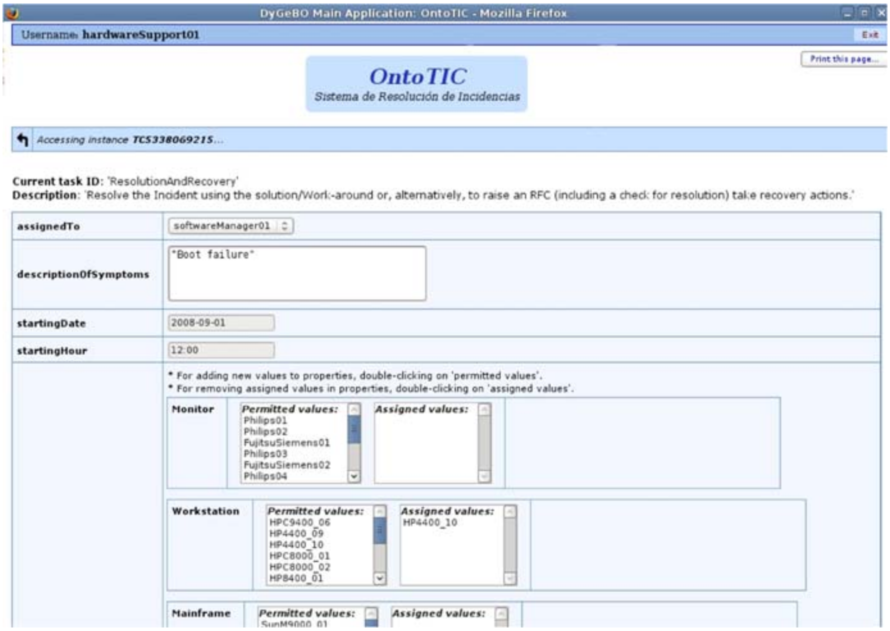
3.7 DyGeBO Tool: Dynamic Generator of WfMS Based on OntoWorkflow

The WfMS provided by our model is a web application called DyGeBO (Dynamic Generator of WfMS based on OntoWorkflow). DyGeBO manages the administrative process represented in the ontologies built using elements and rules defined in OntoWorkflow, i.e., OntoTerms and OntoProcess. Basically, once the participant is identified and has chosen one of the instances of case assigned to him, DyGeBO generates a web form which, depending on the states of a case, will provide the necessary information to perform the task and will show the data fields to be filled in. Therefore, the system is built dynamically from the information specified in the three ontologies. This means that if users want to change either the sequence of tasks, the elements involved in each one of them or the categories of participants required by the tasks, they simply have to update the corresponding ontology and those changes will be automatically transferred to the system. Figure 6 shows DyGeBO tool in incident management domain.

4 Conclusions

Workflows have usually been used to define and manage business process models, where company or institution managers define these processes from scratch or

[8] http://jena.sourceforge.net/.


DyGeBO Main Application: OntoTIC - Mozilla Firefox
Exit
Print this page...
OntoTIC
Sistema de Resolución de Incidencias

Username: **hardwareSupport01**

Accessing instance ***TC5338069215***...

Current task ID: 'ResolutionAndRecovery'
Description: 'Resolve the Incident using the solution/Work-around or, alternatively, to raise an RFC (including a check for resolution) take recovery actions.'

assignedTo	softwareManager01
descriptionOfSymptoms	"Boot failure"
startingDate	2008-09-01
startingHour	12:00

- For adding new values to properties, double-clicking on 'permitted values'.
- For removing assigned values in properties, double-clicking on 'assigned values'.

Monitor	***Permitted values:*** Philips01 Philips02 FujitsuSiemens01 Philips03 FujitsuSiemens02 Philips04	***Assigned values:***
Workstation	***Permitted values:*** HPC9400_06 HP4400_09 HP4400_10 HPC8000_01 HPC8000_02 HP8400_01	***Assigned values:*** HP4400_10
Mainframe	***Permitted values:*** SunM9000_01	***Assigned values:***

Fig. 6 DyGeBO snapshot with an incident management example

reusing other similar models from their organization. There are business processes related to administrative management that do not need computational calculations or invocation of external applications; tasks consist only in the corresponding users providing information from one task to another.

This work shows how the application of ontologies offers significant advantages for this kind of processes, in which the weight of the system lies in the classification of the managed data in the tasks and the categorization of the participant users for each task. These advantages are due to the properties of this type of knowledge representation: ease of use, comprehensiveness, consistency and shared information.

Moreover, the use of ontologies for the representation of workflow processes promotes the reuse, adaptation and integration of the processes and the used data in each task.

To incorporate the advantages of using ontologies in the field of workflows, this paper describes how to represent both process workflows and the case data and users involved in the tasks using ontologies. The elements and rules that define workflows according to the standards and recommendations of the WfMC, are specified in an ontology in the domain of workflows, called OntoWorkflow.

We have implemented a software tool PEBO (PetriNet Editor Based-on OntoWorkflow) to define workflows with the elements and restrictions described in OntoWorkflow. PEBO, using Petri nets, allows users to define intuitively the execution sequence of their business process, data that should appear in every phase, and users responsible for filling in the corresponding information. From ontologies of terms, PEBO allows the user to select this case data and to choose those users responsible for the task.

The processes represented following this model, are not only useful for organizing a company's workflows, but allow them to be executed directly by dynamic Web tools. Thus, the model provides a method that determines the steps and rules that should be followed in order to use the workflows directly in Web applications. That is, this method allows the workflows represented in OntoProcess and OntoTerms ontologies to be exploited using dynamically generated web forms. Based on the proposed method, the DyGeBO software tool (Dynamic Generator Based-on OntoWorkflow) has been implemented. At present, this model has been used by the computer services company MPG Extremadura[9] in the domain of information technology in a project of incident management based on ITIL. Feedback from users, who have utilized the model, reveals that initially, design of workflows with PEBO is complex; however, they state that, once the workflow has been defined and is managed by DyGeBO, it is easy to reuse and adapt for different users.

The definition, merging and reusing of case data from workflows represented in ontologies are relatively simple processes with existing ontology software tools. In contrast, although the ontology merging methodologies can be extrapolated to the processes of workflows represented in ontologies, work has not been found in this field and, therefore, will be a future line of the work presented here. On the other hand, the implementation of transformation modules will be developed from business models (represented in language standards for representation of workflows [21–24]) to ontologies. In this way, methods of reuse and merging of ontologies could be used applied to existing business processes that are currently represented in standard workflow languages. Another future line of work is the inclusion in OntoWorkflow of more complete organisational management.

Acknowledgments This work has been developed under support of Junta de Extremadura Project (PDT08A023) and Ministerio de Ciencia e Innovacion Project (TIN2008-02985). We thank Javier Prieto of MPG Extremadura for his collaboration in the design and implementation of methods and tools.

References

1. Workflow Management Coalition: Workflow management coalition terminology & glossary. Document No. WFMC-TC-1011,. 3rd ed, WfMC, Winchester (1999)
2. van der Aalst, W., van Hee, K.: Workflow management - models, methods and systems, 1st edn. Advanced Information and Knowledge Processing/MIT Press, Cambridge, MA/London (2002)
3. Gartner Group: Business process and business intelligence: The 2005 Focus. Delivering IT's contribution: the 2005 CIO agenda. Gartner, Inc., Stamford, Section 3, pp. 40–51 (2005)
4. McCready, S.: There is more than one kind of workflow software, Computerworld, November 2 (1992)
5. Georgakopoulos, D., Hornick, M.F., Sheth, A.P.: An overview of workflow management: from process modeling to workflow automation infrastructure. Distrib Parallel Databases **3**(2), 119–153 (1995)
6. Weske, M.: Business process management: concepts, languages, architectures. Springer, New York (2007)
7. Vieira, T. A. S. C., Casanova, M. A., Ferrao, L. G.: An ontology-driven architecture for flexible workflow execution. In: Proceedings of the Joint Conference 10th Brazilian Symposium on

[9] http://www.mpgex.es.

Multimedia and the Web & 2nd Latin American Web Congress, IEEE Computer Society, Ribeirao Preto-SP, Brazil, pp. 70–77 (2004)

8. Pathak, J., Caragea, D., Honavar, V.:Ontology-extended component-based workflows: a framework for constructing complex workflows from semantically heterogeneous software components. In: Proceedings of the SWDB Semantic Web and Databases, Second International Workshop, Springer, Toronto, Canada, pp. 41–56 (2004)
9. Zdravkovic, J., Kabilan, V.: Enabling business process interoperability using contract workflow models. In: On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, OTM Confederated International Conferences CoopIS, DOA, and ODBASE 2005 Proceedings, Part I, Springer, Agia Napa, Cyprus, pp. 77–93 (2005)
10. Gasevic, D., Devedzic, V.: Petri net ontology. Knowl.-Based Syst. **19**(4), 220–234 (2006)
11. Haller, A., Oren, E., Kotinurmi, P.: m3po: an ontology to relate choreography models to workflow models. In: Proceedings of the International Conference on Services Computing, IEEE Computer Society, Chicago, Illinois (2006)
12. Vidal, J. C., Lama, M., Bugarín, A.: A workflow modeling framework enhanced with problem-solving knowledge, In: Proceedings of the 10th International Conference on Knowledge-based intelligent information and engineering systems, Springer, Bournemouth, UK, pp. 623–632 (2006)
13. Yao, Z., Liu, S., Han, L., Reddy, Y. V. R., Yu, J., Liu, Y., Zhang, C., Zheng, Z.: An ontology based workflow centric collaboration system. In: Proceedings of the Computer Supported Cooperative Work in Design III, 10th International Conference, CSCWD 2006, Springer, Nanjing, China, pp. 689–698 (2007)
14. Abramowicz, W., Filipowska, A., Kaczmarek, M., Kaczmarek, T.: Semantically enhanced business process modelling notation.In: Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM-2007), CEUR-WS, Innsbruck, Austria, pp. 88–91 (2007)
15. OMG: Business process modeling notation (BPMN), Version 1.0. OMG Final Adopted Specification, OMG, Needham (2006)
16. Andonoff, E., Bouaziz, W., Hanachi, C.: A protocol ontology for inter-organizational workflow coordination. In: Proceedings of the 11th East-European Conference on Advances in Databases and Information Systems (ADBIS), Springer, Varna, Bulgaria, pp. 28–40 (2007)
17. Davies, J., van Harmelen, F., Fensel, D.: Towards the semantic web: ontology-driven knowledge management. John Wiley & Sons, Inc., New York (2002)
18. Gruber, T. R.: Toward principles for the design of ontologies used for knowledge sharing, Technical report KSL 93-04, Knowledge Systems Laboratory, Stanford University, Stanford, (1993)
19. Dou, D., LePendu, P.: Ontology-based integration for relational databases. In: Proceedings of the 2006 ACM Symposium on Applied Computing, ACM, Dijon, France, pp. 461–466 (2006)
20. Grüninger, M., Atefi, K., Fox, M.S.: Ontologies to support process integration in enterprise engineering. Comput. Math. Organ. Theory **6**(4), 381–394 (2000)
21. BEA, IBM, Microsoft, SAP and Siebel: Business Process Execution Language for Web Services Version 1.1 (2003)
22. WfMC: Workflow Standard, Process Definition Interface – XML Process Definition Language, Technical Report Document Number WFMC-TC-1025, Workflow Management Coalition, Winchester (2005)
23. van der Aalst, W., ter Hofstede, A.: YAWL: yet another workflow language. Inf. Syst. **30**, 245–275 (2005)
24. Business Process Management Initiative: Business process modeling language. Specification Version 1.0, BPMI.org, Needham, (2002)
25. Lozano-Tello, A., Gómez-Pérez, A.: Applying the ONTOMETRIC method to measure the suitability of ontologies. In: Green, P., Rosemann, M. (eds.) Business Systems Analysis with Ontologies, pp. 249–269. Idea Group Publising, Hershey, Chapter IX. (2005)
26. Gomez-Perez, A., Corcho, O., Fernandez-Lopez, M.: Ontological engineering: with examples from the areas of knowledge management, e-commerce and the semantic web, 2nd edn. Advanced Information and Knowledge Processing Series/Springer, New York (2007)

Author Biographies

Álvaro E. Prieto is a teaching/research assistant professor of Computer Science at the University of Extremadura, Spain. He has an MSc in Computer Science from the University of Extremadura (2000).

His Ph.D. research addresses the use of ontologies in workflows. He is currently involved in various national and regional R&D&I projects.

Adolfo Lozano-Tello is teaching/research assistant professor of Computer Science Department at University of Extremadura, Spain. He is a Ph.D. (2002) with a special prize of extraordinary thesis about selection of ontologies for software applications. He has published more than 50 papers on the above issues on Software Engineering and Knowledge Engineering.