

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/247928294>

Thesis proposal on "Requirement Engineering Process for Service Oriented Software Development"

Conference Paper · June 2010

DOI: 10.1145/1961258.1961278

CITATIONS

2

READS

3,134

3 authors:



Muneera Bano

Deakin University

55 PUBLICATIONS 842 CITATIONS

[SEE PROFILE](#)



Naveed Ikram

Riphah International University

59 PUBLICATIONS 636 CITATIONS

[SEE PROFILE](#)



Mahmood Niazi

King Fahd University of Petroleum and Minerals

124 PUBLICATIONS 5,187 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Communication and Coordination Challenges Mitigation Model [View project](#)



Towards a Model for Software Process Improvement Implementation and Management in Global Software Development [View project](#)

Thesis Proposal on “Requirement Engineering Process for Service Oriented Software Development”

Muneera Bano
Department of Software Engineering
International Islamic University
Islamabad, Pakistan
muneera@iiu.edu.pk

Supervisor: Naveed Ikram
Department of Software Engineering
International Islamic University
Islamabad, Pakistan
naveed.ikram@iiu.edu.pk

Supervisor: Mahmood Niazi
School of Computing and
Mathematics, Keele University, UK
m.k.niazi@cs.keel.ac.uk

ABSTRACT

Service Oriented Architecture (SOA) has become the new reference architecture for distributed computing. For the last few years the number of services on the web has increased exponentially. Among available services locating the best service that fulfills the user requirement is a challenging task for researchers especially when they are emphasizing on the need of automating the process of web service discovery. There is still no standard Requirement Engineering Process defined for service centric systems, as the traditional processes and those used for COTS selection cannot be used due to the architectural differences of service oriented software development to the other domains. The aim of this thesis proposal is to highlight the issues and challenges for service oriented software development paradigms and then to create a framework based on the solutions to the issues and then validating this proposal with the help of experiments.

Categories and Subject Descriptors

D.2.1 [Requirements/Specifications]: Methodologies

General Terms

Design, Experimentation, Standardization.

Keywords

Requirement Engineering, Service Oriented Software Development, Component based Software Development.

1. INTRODUCTION

Component Based Software Development has emerged as a field of software engineering to make use of available software components to provide the facility of reusability and reducing the effort for effective management of complexity and raising competition in market for enhancing the quality of components [1]. The internet has taken an important role in the field of computing and global communication so using it for distributed computing looks promising as it provides appropriate infrastructure and technologies for distributed application development where developers is provided with transparency to the underlying working technologies and they focus on designing application by locating the required components in distributed environment. This view takes us to the new paradigm of software engineering known as Service Oriented Software Development [2]. The software applications in this paradigm are based on web service usage which are software components available on the internet with web interfaces and can be leased to be used as a part of software. SOSD can be seen as evolution of CBSD [3] by providing the new architectural style for building applications that support loose coupling among web services. These services can

be located in central repository UDDI and after service level agreement (SLA) can be used by composing an application. Fig 1 shows the evolutionary link among traditional software development methods to component based software development and then ultimately towards service oriented software development.



Fig 1: The evolutionary Link

The two paradigms of component based software engineering and service oriented software engineering are trying to come up with solutions to the challenges posed by large and complex system designing and implementation. CBSD though proved promising for software reuse and maintainability but it still faces issues like heterogeneity of platforms and protocols and difficulty of locating required components and selecting them against system requirements. So SOSD is now the emerging paradigm that seems to provide solution to the problems faced in CBSD so as to make application development easy, low cost even in heterogeneous distributed environment based web services which can easily be located over the internet.

Currently the research community is dealing with various issues and challenged posed by this new field, those areas that are concerned with requirement engineering are [3];

- Service composition
The focus here is on the methods and techniques for discovering and selecting the web services based on system requirements and then integrating them by any orchestrating language or tool to compose an application. The main challenge occurs in dynamic when multiple services are available and a dynamic criterion for selection is required.
- Service management and monitoring
This phase deals with managing and controlling the service based applications throughout the life cycle; When some services are no longer available which are being used as a part of composed service, Requirements of existing system evolves and re-composition is required, New version of web service is launched by the provider which raises conflicts with existing system requirements

Software is considered successful if it meets the objective for which it was made and to identify that objective we use Requirement Engineering process [4]. The process includes; [5]

- Identification of stakeholders
- Gathering their needs and understanding them
- Documenting the specifications
- Analyzing it for subsequent implementation

RE can be seen as made up of two phases;

- Requirement Development; It includes activities like, Requirement Elicitation, Analysis, Specification and Verification
- Requirement Management; It deals with managing and controlling changing and evolving requirements

The result of requirement development should be an agreement of stakeholders about the system to be built and during requirement management the changes to those agreed requirements are controlled and managed.

As Service based Software Development is a shift from traditional development paradigm, several new methods have been proposed for it. RE phase is also affected by the SDLC used for development [4]. For example in traditional waterfall model, major effort in RE phase is to acquire, specify and analyze the requirements during analysis phase but in case of agile process, requirements are taken only in informal scenarios as the main focus here is on coding phase. So the way in which the requirements are gathered, specified and used is different in various development life cycles. Keeping the above discussed view in mind we can clearly see that SOSD is a different architectural style the traditional development styles for software applications and hence requires a complete new RE process [6] considering only the service oriented paradigm of software development life cycle [3], [2]. RE for SOSD can involve activities of traditional process like modeling, specification, and analysis but the way in which these activities are carried out is different. The main focus here is to identify the services that match the system requirements and then modeling a composition on the basis of selected services. Based on system's current status or changing user requirements a SOA application can dynamically change its services, workflows, web interfaces etc. The architecture [7] on which the service based systems are built is producer centric where service provider publishes the specification of the service to be launched in the central registry UDDI. And the users locate them using WSDL (XML based messaging language) according to the system needs and then using them as a part of their application after service level agreement. The central registry here can play an important role by providing the service providers with specifications based on user's queries and let the supplier provide a service according to the demand from users. These services can be treated as prototypes which can be used for improving the incomplete specifications provided by users. The service providers also create requirements based on current market trends and thus bringing creativity and innovation as a part of service oriented development [8], [9], [10]. Still there has been no standard for following any RE process in this domain.

2. RESEARCH OBJECTIVE

The research objectives of this thesis proposal are;

- To provide a framework based on solution to those issues for RE process in service oriented systems
- Validating the proposed framework with help of experiment by scenario creation

3. LITERATURE REVIEW

As we can see that the SOSD is closer in nature to CBSD as they hold many similarities [7] that means we will have to look into the issues of RE that are faced by CBSD to see if they are inherited to the SOSD while keeping in view what the original issues of RE are. The following three categories show the

challenges of RE in traditional software development, in CBSD and in SOSD.

3.1 Issues and Challenges of RE in traditional Software Development

RE is a challenging task and the success of any project depends heavily on the activities carried out in this phase. The main issues that are faced by researchers working in this field are;

- Issues regarding stakeholders [11], [12], [13]
- Capturing, modeling and analyzing Non Functional Requirements[13]
- Reuse of requirement models [14]
- Formal representation of requirements from natural language[15]
- Requirement change and evolution
- Conflict resolution in requirements[12]
- Creating requirements [8]

3.2 Issues and Challenges of RE in component based software development

CBSD Life cycle is different from traditional that means exact RE process cannot be applied here and some new methods and techniques are required. CBSD life cycle follows steps as; requirement analysis, software architecture selection, construction, analysis and evaluation, components identification, selection and customization, system integration, testing and maintenance [16], [17]. So the challenges of RE here differ from those present in the traditional software development. [18]

- No standard RE process exists for CBSD, specially for searching and selection of components [16], [19]
- NFR can play an important role in Quality comparison among multiple components providing same functionality. As components have black box nature they can be only tested against user criteria for quality. [16], [17], [20], [21], [22]
- Specifications of existing components are also to be considered when building new system's requirements. Traditional approaches cannot be applied here. [20]
- Search and selection process is the most important phase in CBSD life cycle [20]
- Systematic evaluation and testing of components is required against user requirements [21]
- Tradeoff is required between user requirements and component selection, so a flexible and iterative RE technique is required for making decisions for component selection. [20], [21], [22], [23]
- Incompatible components can fail the system at the time of integration [22]
- As components are black box in their nature [20], [21] the source code is not provided and they are inflexible for customization. [22]
- Versioning in components can cause problems as new version may not match the existing system requirements [22]
- The process should be an iterative and should be able to do knowledge sharing which is supply chain of components/products, skills and experiences and personnel.[19], [24]

3.3 Issues and Challenges of RE in service oriented software development

The issues in the literature on which the researchers focus are to some extent inherited from CBSD along with those issues that were present in traditional software development, but some are specific to SOSD. Based on the comparison with CBSD we have gathered following issues;

- Service Discovery is the most important phase so effective mechanisms are required to locate correct service according to user requirements [25]

- Automated to support dynamic service discovery based on user requirements specifications [25]
- An iterative process to refine the requirement specifications [26], [27]
- Using web service discovery as a process to complete the requirements [26], [27]
- Supporting high level language support for requirement specification
- Having innovative and creative requirement phase as integral part of the RE process [8], [9], [10]
- Able to redesign and redeploy the composed service when user needs change over time
- Bridge the semantic gaps which are inevitable when services are brought together from hybrid environments
- Able to manage the knowledge of group/cluster of services with similar functionality
- Apply user NFR for selection among multiple available candidate services
- Able to give orchestration mechanism for service composition and avoid deadlocks in service invocation

4. PROBLEM ANALYSIS

The issues identified above fall into four categories; some of them overlap in different categories making them interlinked to one another. They are;

4.1 Specification issues

By specification issues we actually mean what the actual application that we are going to build should be like, what are its requirements and what services are required that will fulfill these requirements, how we will get these requirements and how we will make them complete. Following issues fall in this category from the set of issues identifies above;

- Support of high level language for specifying initial requirements and then converting them to formal specification
- Creating requirements and making the process innovative to compete in market
- Bridging the semantic gaps which are the result of heterogeneous environments
- Knowledge management of previously completed specification so they can be used for system update (service re-composition) when requirements evolve or new version of service is available
- Discovering web service dependencies on each other from the messages by which they communicate with each other
- Making the discovery process iterative to make specifications complete by treating the resulted services as prototypes

4.2 Service discovery issues

This category deals with searching for the services after the specifications are complete and finding out which of the services actually meet the functional and non functional requirements. They include issues as;

- Automated and dynamic (runtime) discovery issues
- Query generation from formal specification
- NFR and QA testing on resulted services to check if they meet the quality criteria
- Making the discovery process iterative to make specifications complete by treating the resulted services as prototypes

4.3 Knowledge management issues

We would need some knowledge of previous composition of the knowledge about the functionalities of services, so that it would

help in both specification and discovery. The issues of knowledge management from above identified issues are;

- Knowledge management of previously completed specification so they can be used for system update (service re-composition) when requirements evolve or new version of service is available
- Clustering the specification of services according to their functionality by putting same functionality services in same category to reduce the search efforts and increase domain knowledge

4.4 Composition issues

When services are discovered based on their individual functionalities then we need to see if they will work properly in a workflow by making composition and see if the integrated system meets the original requirements. Issues here can be;

- Deadlock management in composition when services depend on each other for completing their functionality
- Re-deploy and re-design the composition when requirements evolve or new version of service is available

5. WORK DONE SO FAR

SOSD is currently under consideration of research community from different perspectives. There have been many methods, techniques and tools proposed by different mega projects and research teams.

SeCSE [28]: Service Centric System Engineering project had the aim to create free and open source methods, tools and techniques for system integrators and service providers to support the cost effective development and use of dependable services and service centric applications. They have incorporated RE with three phases of service discovery; Early Service Discovery, Design time Discovery and Run Time Discovery.

SENSORIA [29]: "Software Engineering for Service Oriented Overlay Computers" had the aim to develop a new approach for service oriented software engineering with foundation theories, techniques and methods. Their focus is on whole SDLC in Service oriented paradigm, from requirements to deployment including re-engineering of legacy systems. The proposed methods and tools make use of mathematical theories and models for ensuring correctness of the procedure and allowing a semi-automatic design process.

IBM SOA [30]: IBM has proposed Service Oriented Modeling and Architecture (SOMA) [31] which is an end-to-end software development method for building SOA based solutions. They provide guidance on identification of services just like objects are identified in Object Oriented Software Development.

SORE workshop [33]: Service Oriented Requirement Engineering workshops were held in 2004 and 2007. Their aim was to gather the research community and share the ideas, knowledge and work on requirement engineering for service oriented systems.

Other Researchers: In [2] Tsai et.al have discussed Service Oriented Software Engineering. In [6] they have focused on RE phase of SOSE and proposed a SORE framework. Xiang et.al [32] have proposed SREM (Service Requirement Elicitation Mechanism) which is based on SRMO, which has taken basic concepts from agent-oriented modeling framework i*. Value based software development concepts are used in Value Gap Model [34] for eliciting requirements for service components.

6. PROPOSED SOLUTION

Based on the issues identified, the first step towards the solution is identification of a framework for Requirement Engineering Process which will consider solutions to these issues. The important step would be to find out how these problems impact and their solutions impact other problems and solutions. Second part would be creating experimental scenarios by which the developed framework would be validated. In the experiment various aspects would be validated from different point of views, e.g. service provider and service consumer.

7. REFERENCES

- [1] I. Crnkovic, "Component-based software engineering-new challenges in software development," *Software Focus*, vol. 2, 2001, pp. 127-133.
- [2] W.T. Tsai, "Service-oriented system engineering: a new paradigm," *IEEE international workshop on service-oriented system engineering (SOSE), Beijing*, 2005, pp. 3-8.
- [3] H.P. Breivold and M. Larsson, "Component-based and service-oriented software engineering: Key concepts and principles," *SOFTWARE ARCHITECTURE EVOLUTION AND SOFTWARE EVOLVABILITY*, p. 67.
- [4] B. Nuseibeh and S. Easterbrook, "Requirements engineering: a roadmap," *Proceedings of the Conference on the Future of Software Engineering*, ACM New York, NY, USA, 2000, pp. 35-46.
- [5] D.E.H. Damian, "Challenges in Requirements Engineering," *Requirements E, Springer, Springer*, vol. 8, 2003, pp. 149-160.
- [6] W.T. Tsai, Z. Jin, P. Wang, and B. Wu, "Requirement engineering in service-oriented system engineering," *proceedings of International Workshop on Service-Oriented System Engineering*, Citeseer, , pp. 661-668.
- [7] A. Brown, S. Johnston, and K. Kelly, "Using service-oriented architecture and component-based development to build web service applications," *interactions*, vol. 1, p. 2.
- [8] N. Maiden, S. Robertson, and J. Robertson, "Creative requirements: invention and its role in requirements engineering," *Proceedings of the 28th international conference on Software engineering*, ACM, 2006, p. 1074.
- [9] N. Maiden, "Servicing your requirements," *IEEE software*, 2006, pp. 14-16.
- [10] N. Maiden, A. Gizikis, and S. Robertson, "Provoking creativity: Imagine what your requirements could be like," *IEEE software*, vol. 21, 2004, pp. 68-75.
- [11] B. Regnell, M. Höst, J.N. och Dag, P. Beremark, and T. Hjelm, "An industrial case study on distributed prioritisation in market-driven requirements engineering for packaged software," *Requirements Engineering*, vol. 6, 2001, pp. 51-62.
- [12] G. Spanoudakis and A. Zisman, "Inconsistency management in software engineering: Survey and open research issues," *Handbook of software engineering and knowledge engineering*, vol. 1, 2001, pp. 24-29.
- [13] B. Decker, E. Ras, J. Rech, P. Jaubert, and M. Rieth, "Wiki-based stakeholder participation in requirements engineering," *IEEE software*, 2007, pp. 28-35.
- [14] B. Nuseibeh and S. Easterbrook, "Requirements engineering: a roadmap," *Proceedings of the Conference on the Future of Software Engineering*, ACM New York, NY, USA, 2000, pp. 35-46.
- [15] H. Kaindl, "The missing link in requirements engineering," *ACM SIGSOFT Software Engineering Notes*, vol. 18, 1993, pp. 30-39.
- [16] X. Cai, M.R. Lyu, K.F. Wong, and R. Ko, "Component-based software engineering: technologies, development frameworks, and quality assurance schemes," *Proc. Asia-Pacific Software Engineering Conf*, 2000, pp. 372-379.
- [17] L. Iribarne, A. Vallecillo, C. Alves, and J. Castro, "A non-functional approach for COTS components trading," *Proc. of WER*, Citeseer, 2001, pp. 124-138.
- [18] L. Karlsson, Å. Dahlstedt, J.N. och Dag, B. Regnell, and A. Persson, "Challenges in market-driven requirements engineering-an industrial interview study," *Proceedings of the Eighth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'02)*, Citeseer, 2003, pp. 101-112.
- [19] C. Ncube and N.A.M. Maiden, "PORE: Procurement-oriented requirements engineering method for the component-based systems engineering development paradigm," *International Workshop on Component-Based Software Engineering*, Citeseer, 1999.
- [20] L. Iribarne, J.M. Troya, and A. Vallecillo, "A trading service for COTS components," *The Computer Journal*, vol. 47, 2004, p. 342.
- [21] S. Ghosh, J.L. Kelly, and R.P. Shankar, "Enabling the Selection of COTS Components," *COTS-Based Software Systems, LNCS*, vol. 3412, 2005.
- [22] W. Kozaczynski and G. Booch, "Component-based software engineering," *IEEE software*, vol. 15, 1998, pp. 34-36.
- [23] C. Alves and A. Finkelstein, "Challenges in COTS decision-making: a goal-driven requirements engineering perspective," *Proceedings of the 14th international conference on Software engineering and knowledge engineering*, ACM New York, NY, USA, 2002, pp. 789-794.
- [24] L. Chung, K. Cooper, and S. Courtney, "COTS-Aware Requirements Engineering: The CARE Process," *Proc. of Int. Workshop on Requirements Engineering on COTS (RECOTS '04)*, 2004.
- [25] G. Spanoudakis, A. Zisman, and A. Kozlenkov, "A service discovery framework for service centric systems," *Proceedings of the IEEE International Conference on Services Computing (SCC'05)*, Citeseer, 2005, pp. 251-259.
- [26] K. Zachos, N. Maiden, X. Zhu, and S. Jones, "Discovering web services to specify more complete system requirements," *Lecture Notes in Computer Science*, vol. 4495, 2007, p. 142.
- [27] K. Zachos, N. Maiden, X. Zhu, and S. Jones, "Does Service Discovery Enhance Requirements Specification? A Preliminary Empirical Investigation," *Service-Oriented Computing: Consequences for Engineering Requirements, 2006. SOCCER'06*, 2006, pp. 2-2.
- [28] S. Crew, *Service Centric System Engineering—EU/IST Integrated Project*, 2004.
- [29] M. Wirsing and M. Hölzl, *Rigorous Software Engineering for Service-Oriented Systems—Results of the Sensoria project on Software Engineering for Service-Oriented Computing*, Springer, 2010.
- [30] A. Arsanjani, L.J. Zhang, M. Ellis, A. Allam, and K. Channabasavaiah, "S3: A service-oriented reference architecture," *IT professional*, vol. 9, 2007, pp. 10-17.
- [31] A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Ganapathy, and K. Holley, "SOMA: A method for developing service-oriented solutions," *IBM systems Journal*, vol. 47, 2008, pp. 377-396.
- [32] J. Xiang, L. Liu, W. Qiao, and J. Yang, "SREM: A Service Requirements Elicitation Mechanism based on Ontology," *Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International*, 2007.
- [33] <http://conferenze.dei.polimi.it/sore04/>
- [34] Sang Won Lim, Taek Lee, Sangsoo Kim, Hoh Peter In, "The Value Gap Model: Value-Based Requirements Elicitation," *Computer and Information Technology, International Conference on*, pp. 885-890, 7th IEEE International Conference on Computer and Information Technology (CIT 2007), 2007