2012 International Conference on Networks and Information (ICNI 2012)
IPCSIT vol. 57 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V57.19

Requirement Engineering in Service-Oriented Architecture

Atefeh Khosravi [1+], Nasser Modiri[2]

[1] MS student of software engineering,Islamic Azad University of Tehran Northern Branch- Tehran, Iran

[2] Lecturer at department of computer engineering, Islamic Azad University of Zanjan- Zanjan, Iran

Abstract. Service-Oriented Architecture (SOA) becomes more popular recently and loads of applications are developed using this approach. There are two main groups according to this architecture, service providers and service consumers.
The main point of SOA is to provide the ability that service consumers can develop their applications using services provided by service providers. But here the crucial issue is that services should be accurate and in accordance of consumers' requirements. Services should be designed reusable so that development of new applications becomes faster and cost of it will decrease. So specifying requirements accurately and completely is essential and the necessity of requirement engineering emerges here.
In this article we explain the vitality of RE (Requirement Engineering) in SOA and will propose a guideline to implement RE in service-oriented environments.

Keywords: Requirement engineering, RE, Service oriented requirement engineering, SORE

1. Introduction

Service-Oriented Architecture (SOA) receives significant attention recently and as cloud computing and cloud marketing is growing, SOA attracts more consideration. Designing services according to customer's requirement is very important in this architecture. Services should be independent and reusable, to do so we need to know the exact goal of the system. We can utilize requirement engineering to specify goals and requirements to elicit services.

In fact, requirement engineering has evolved from classical methods such as SREM, to object oriented methods such as UML and finally to Service-Oriented Requirement Engineering (SORE) [1].

In this paper we want to show the importance of requirements in service elicitation in service oriented architecture and proposed our applied guideline to specify services more accurate. In the next section we study classic RE in brief. In section 3 we focus on key features of SOA and then in section 4 we discuss SORE. And finally we propose a guideline to support SORE.

2. Classic RE

System requirements are the description of functionalities which it is expected to be supplied. These requirements are customers' organization goals and they should be satisfied by application [2].

We can classify requirements into two main groups: User requirements and system requirements.

User requirements should specify functional and non-functional requirements empty of jargons and easy to understand for users. It should define external behavior of system and avoid details of system design.

System requirements are developed version of user requirements which can be used as the starting point for designing system. They contain system details and describe how user requirements should be supported.

[+] *E-mail address:* ati.khosravi@gmail.com; nassermodiri@yahoo.com

These requirements should be specified accurate and complete, and for decreasing ambiguity it is better to describe them by using natural and structured languages which are capable of containing tables and system models.

Requirement engineering discovers, models and specifies and documents these requirements for a system and its domain.

Classic RE has four phases: feasibility study, requirement elicitation and analyze, requirement specification and requirement validation. Generally RE specifies that whether the system is convenient for the desired business and if so it discovers requirements and standardizes them. And finally in validation phase it evaluates that if requirements are what customers want [2, 3].

2.1. Feasibility study

For all new systems RE process must start from feasibility study. Primary business requirements, system general description and the way that the system is going to support the business are inputs for this phase. The outcome will be a report to make a decision base on, to determine if development of the system would be fair or not.

2.2. Requirements elicitation and analyze

In this phase engineers contact customers and end users to collect information about system domain, system functionalities, services should be provided, system constraints and etc. Requirements elicitation and analyze may cause involvement of different stockholder with different role.

Figure 1 shows requirement elicitation and analyze during requirement engineering.

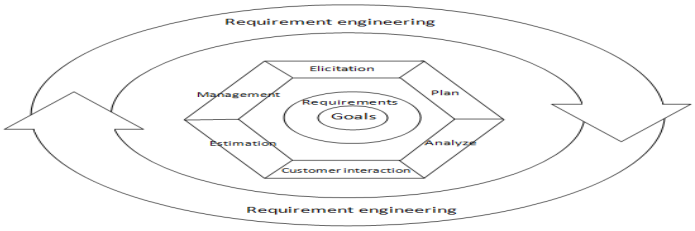

Fig. 1.Requirement elicitation in RE process

2.3. Requirement specification

The goal of requirement specification is to achieve a common understanding of requirements. In this phase requirements have to be documented in different levels for usage of different users. For example documents which are used by developers should contain detail of system and technical issues to provide them. However, there is no need to put technical solution for in a document used by testers.

Requirement documents describe system and it domain. They can be used as criteria for future processes such as system detail design, test cases, validation and verification and change management.

2.4. Requirement validation

This phase examines that whether supplied functionalities are relevant to customer's requirements or not. This phase is a bit overlapped with requirements analyze in the matter that validation phase focuses on discovering problems and conflicts among requirements and checks their compatibility.

This phase is very important because when an undiscovered problem becomes detected in developing phase or after that, the cost of its correction will be huge.

3. SOA

The main idea of SOA is based on composition of object oriented architecture and component base architecture [4].

In this architecture developers are divided into three independent but cooperative groups: Application builders or services clients, service brokers and service providers.

Service providers' task is to provide independent and loosely coupled services.

Service brokers' duty is service introduction and marketing. Application builders find their required services for constructing their applications via service brokers.

So that services can be placed in center of SOA. In this way service providers supply some services and application builders can select published services through brokers to construct an application. However, SOA can be consumer centric. In this way application builders announce their requirements and service providers supply them by services [5].

In fact the most important goal of SOA is to provide services which are exactly what consumers want and are as much independent and reusable as possible. And without depending on specific platforms they can make development process faster so that the cost of development decreases. Services should support activities in business process. So in SOA business processes should be specified firstly, then after identifying activities required for each of them we should decide for which of these activities services should be designed. It is possible that designer decide to design several services for an activity because some part of this activity may be used in other business processes. Or it is possible to design one service for several activities. Despite the designer approach in designing services, it is important for him or her to keep major feature of services (independency and reusability) in him or her mind.

Figure 2 demonstrates the position of service consumers and service providers, and the relation between business processes and services.

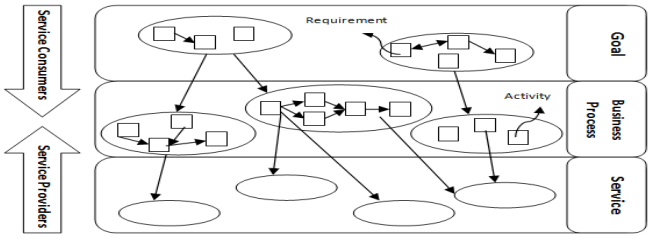

Fig. 2.Relation between business processes and services

4. Requirement engineering in service oriented architecture

The main principle of developing an application is to be based on customers' requirements. Specifying and supplying customers' requirements have been the subjects which developers are worried about. Supplying requirements are varied in different software life cycle. For example in linear approach the first phase is devoted to requirements specification, while in spiral models in each iteration customer's requirements should be considered. But as RE shows, RE is not specific for a single phase, it is an umbrella activity which covers all phases involved in product life cycle.

In organizations which are base on it services, there are two major groups involving with services [6]:

- Users who are responsible for business functionality and they need IT support for it, and IT support them by providing services,
- And people who are in IT department working on these services design and implementation. This group always attempt to providing services which are presentable to several supplicants or useable for supplying common requirements among different business processes [7].

However, utilizing existing services is still an issue for many organizations because simply services cannot supply their requirements [8]. The main reason is that services are designed too specific with limited observation and this causes more constraints on reusing services. On the other hand in defining new projects, potential capabilities of existing services become neglected and while existing services can supply some requirement new services are designed for business processes. It means that reusability of existing services is ignored by project team.

Requirement engineering in SOA is an interface between service engineering and application engineering [9]. In fact Service-Oriented RE (SORE) is applying classic RE for both group of service consumers and service suppliers. And considering that SORE is applying in a service-oriented environment with a service-oriented infrastructure, it will be different with classic RE in entities it discovers and the methods it uses to discover entities [5]. Instead of discovering objects and classes, SORE specifies business process services and flows. And the effect of it on decreasing business process change control and management causes SORE to become more important. So that if we design services based on requirements

and ideally, as applications are based on services with small changes in services we can project desired changes through an application.

5. Proposed guideline

To optimizing the effect of requirement engineering on service oriented architecture we propose this guideline:

1. Specifying goals: goals are vital parts in requirement engineering and have a key role on a project succeed or fail. Specifying goals complete and accurate while considering organization policies is difficult but essential to help requirements identification. Sometimes it is possible to divide goals to some sub-goals to make its identification easier.
2. Specifying user's requirement: as we mentioned, requirements can be divided into user's requirements and system requirements. In this step, user's requirements should be specified according to specified goals in previous step. And stockholders should validate and verify them.
3. Specifying business processes: for supplying customers' requirements we should specify business processes and understand their tasks well. This step is very important because software engineers may have limited knowledge about business. Consequently an obvious subject for customers (they don't think it is necessary to explain it) has another meaning to engineers. So that the final product will be based on software engineers personal views. In this step we should model business processes and they should be validated by stockholders.
4. Specifying business process domain: for supplying a business process we should know its activities. To design services, reuse them and change them with less cost, it will be helpful to consider several domains for business processes. For example, in providing services for a hospital and a restaurant, each of them has its own specific business processes in different domains.
   In this step specifying domain of each business process is important because it ends to consider services existing in specified domain or designing services with wider view which is has effect on reusability of services.
5. Dividing domains into subsystems: in this step we divide a domain into subsystems. So that system modularity will be preserved and its management and maintenance will be easier.
6. Specifying convenient subsystem for a service: in this step we should specify convenient subsystem for a service. Categorizing services into suitable subsystem causes logical arrangement and easier system monitoring. Furthermore it helps to make system configuration easier. For instance, imagine that we want to sell a bank application to two different banks. All the functionality is same but one of banks does not have currency exchange system. We can simply disable currency exchange subsystem and it services and there is no need to design a new system for that bank.

Generally as our classification is more efficient and as we have logical abstraction services reusability will increases, change management will be easier and maintenance cost will decreases.

To make our guideline clearer, imagine we want to design an application for a retail store.
Figure 3 demonstrates our applied guideline on this case study.

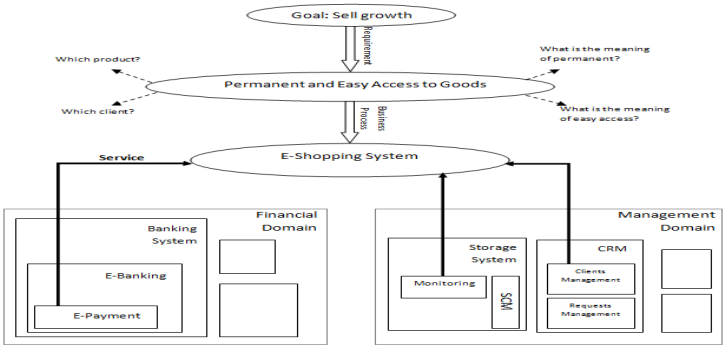

Fig. 3.SORE in an e-shopping system

Goal specification is the first step. According to stockholders the goal is sell growth. And to achieve this goal *easy access* to *goods* for *customers* at *all times* is necessary. According to our guideline in this step we should specify requirements accurate. So that we should discover what stockholder do mean by *easy access, goods, customers* and *always*. One of recommended business processes to achieve this goal is to set up an e-shopping system. After modeling this business process and obtaining the stockholder's agreement, according to its tasks we should specify the domain of this business process. In this process customer should be able to register on site and have its own profile so that there will be no need to insert its address each time. Customer

should know what goods are available and what their specifications are. And finally customer should be able to pay for desired good on line. So this business process uses financial and management domain. In management domain storage system and CRM exist. CRM can contain customer's request management subsystem or customer management subsystem (to hold customers profile data and to recognize valuable customers). According to our business process we should look for convenient services in customer management subsystem and we should implement them there if they don't exist. Storage system can contain monitoring subsystem (to monitoring goods and determining how many of what product is available) or ordering subsystem (to determine how many of what product should be ordered and when ordering date should be when the delivery date is). According to our business process we should look for convenient services in monitoring subsystem and we should implement them there if they don't exist. And finally for online payment we should use services existing in e-banking subsystem.

6. Conclusion

While service-oriented architecture become more popular requirement engineering in service-oriented architecture will be more important. As service classification is more logical and accurate and as we provide modules with minimum dependencies which are cooperating via convenient interfaces, agility in software development increases and the cost of change management and maintenance decrease.

Obviously this goal is achieved only by identifying and specifying services and business processes accurate and complete which is the result of applying requirement engineering in a service-oriented architecture.

7. References

[1] IBM Developers Works. IBM SOA Foundation: An architectural introduction and overview, www.ibm.com/developerworks/webservices/ws-soa-whitepaper

[2] F.Mejia, J.Tavarez, F.Alvarez, L.Gonzalez, H.Limon, H From Requirements Engineering To Service Oriented Requirement Engineering:An Analysis Of Transition", ICIS, 2009

[3] I.Sommerville, Software Engineering, 8th edition, pp.142 –147

[4] W. T. Tsai, Service-Oriented System Engineering: A New Paradigm, IEEE SOSE, 2005

[5] W. T. Tsai, Z. Jin, P. Wang, B. Wu, Requirement Engineering in Service-Oriented System Engineering, IEEE ICEBE, 2007

[6] P.Eck, R.Wieringa, Requirements Engineering for Service-Oriented Computing: A Position Paper, ICEC, 2003

[7] Sh. Lichtenstein, L. Nguyen, A. Hunter, Issues in IT Service-Oriented Requirements Engineering, Australasian Journal of Information Systems, Vol 13, No 1, 2005

[8] S. Adam, J. Doerr, The Role of Service Abstraction and Service Variability and its Impact on Requirements Engineering for Service-oriented Systems, IEEE COMPSAC, 2008

[9] F. Flores, M. Mora, F. Alvarez, L. Garza, H. Duran, Towards a Systematic Service-oriented Requirements Engineering Process (S-SoRE), CENTERIS, 2010

Atefeh Khosravi, born on 1987, IranM.Sc student of software engineering in Islamic Azad University-Tehran Northern Branch (Tehran/Iran). Received her B.Sc in software engineering from Islamic Azad University-Tehran Northern Branch (Tehran/Iran).

Currently she is software analyzer and designer in Tosan LTD in Tehran, developing CoreBanking systems. She is interested in requirement engineering, business process analysis and service oriented computing.

Nasser Modiri, born on 1962, April 21 DOB.Received his M.Sc and PhD in Electronics engineering from the University of Southampton (UK) and the University of Sussex (UK). He is currently, Assistant Professor of Department of Computer Engineering Islamic Azad University (Zanjan/Iran). Research interests include Network Operation Centres, Framework for Securing Networks, Virtual Organizations, RFID and Product Life Cycle Development.