IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661, ISBN: 2278-8727Volume 7, Issue 2 (Nov. - Dec. 2012), PP 01-08
www.iosrjournals.org

Requirement Engineering for Service Based Systems

T.Priyaradhikadevi [1], Dr. R. M. S. Parvathi.[2]

[1,] Research Scholar & Associate Professor , Department of Computer Science and Engineering , Mailam Engineering College, Mailam – 604304, Tamil Nadu, India
[2,] Principal& Professor, Department of Computer Science and Engineering, Sengunthar College of Engineering, Tiruchengode – 637 205, Tamilnadu, India

Abstract: *Requirements engineering for business solutions have dealt with the elicitation, specification, validation, management and change to information systems to be accessed through PCs and workstations. As service based systems have many challenges to access the information. Thus there are many problems in security purpose so we have proposed objectiver approach tool which have been improve the authority. In this paper we also provide the application of online bank service to access the transaction of amount and encryption to user account. This paper summarizes some of the challenges and points to areas for research that should receive the attention of the RE community.*

Keywords- *Encryption, Objective approach tool, Requirement Engineering, Quality, Service*

I. INTRODUCTION

Software Engineering is the establishment and use of software engineering principles in order to obtain economically software that is reliable and works efficiently on real machines. Software Engineering can be treated as a layered technology with requirement, design, and implementation as some of its layers. The goal of software engineering is to produce quality software, which is delivered on time, within budget, and which satisfies the customer’s requirements and user’s needs. A Service approach must rest on an organizational commitment to quality. The bedrock that supports software engineering is a quality focus. The initial phase starts with the gathering of requirements based on the needs of a customer or a user. Design phase specifies how the developed software should perform its tasks. It consists of modularization and detailed interfaces of design elements, their algorithms and procedures, and the data type needed to support the architecture and to satisfy its needs. Testing is made to verify and validate the process before deployment. Qualities can be differentiated as internal and external quality. The external qualities are visible to the user such as reliability, efficiency, usability etc. Whereas the internal quality is concerned with the developers they are verifiability, maintainability, adaptability etc.

The requirements engineering are subject to change and new requirements must be elicited and documented and existing requirements managed over the lifetime of the system. Using Requirement Engineering we can complete project on time, on budget, and with all functions originally specified is to be success of the project and the failure is to cancellation before completion or never be implemented. The challenges are the project to be completed but over budget, late, and with fewer function than initially specified.

Using services as basic concept eases the specification of reactive systems with a high degree of interaction with its environment as found, e.g., in the telecommunication or web services domain. This approach allows breaking up complex system functionality into smaller functional modules. The modularity supports a more manageable and comprehensible description of the functionality. There, systems do not consist of a fixed set of components, but are dynamically composed from services. However, using a service-based engineering process is not only useful in the field of dynamic networks, but also in domains with static structure supplying complex interacting functionalities.

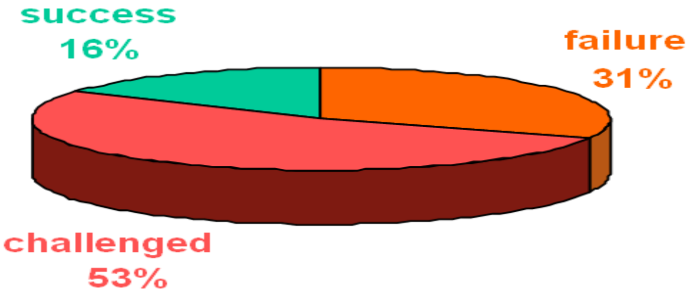


Figure 1. Requirement engineering

II. SCOPE OF THE PROJECT

The scalability issues in Web services may be different from those of traditional programs. Scale isn't so much about lines of code as it is about crossing enterprise boundaries. Some dimensions along with Web services must confront scale include:

- Complexity of the workflow
- Volume of data
- number of nodes
- complexity of operations
- differences in usage patterns

Web services rely on the standardized exchange of data, though standards like WSDL may be more important than lower-level syntactic notations such as XML. Indeed, WSDL files sometimes offer two interfaces to a given service: one, a "standard" interface, that uses an XML notation, and a "customized" interface that uses a domain-specific and more concise, standardized or otherwise appropriate data exchange format.

A Web service exhibits the following defining characteristics:

- A Web service is a Web resource. You access a Web service using platform-independent and language-neutral Web protocols, such as HTTP. These Web protocols ensure easy integration of heterogeneous environments.
- Web services seem to inherently be asynchronous. This helps them better adapt to problems of availability and distribution. This in turn makes them truly different from traditional RPC mechanisms and more like a distributed dataflow notation.
- A Web service provides an interface—a Web API—that can be called from another program. This application-to-application programming interface can be invoked from any type of application. The Web API provides access to the application logic that implements the service.
- Web services rely on the standardized exchange of data, though standards like WSDL may be more important than lower-level syntactic notations such as XML. Indeed, WSDL files sometimes offer two interfaces to a given service: one, a "standard" interface, that uses an XML notation, and a "customized" interface that uses a domain-specific and more concise, standardized or otherwise appropriate data exchange format.
- A Web service is typically registered and can be located through a Web service registry. A registry enables service consumers to find services that match their needs. These service consumers may be humans or other applications.
- Web services support loosely coupled connections between systems. Web services communicate by passing XML messages to each other via a Web API, which adds a layer of abstraction to the environment that makes the connections flexible and adaptable.
- Distribution is an important part of Web services, and this significantly impacts any form of analysis—do you have access to the internals, even the specifications? These problems are especially manifest when systems are no longer within a single enterprise.
- Web services create problems of availability and dynamicity, especially in the context of automated discovery and composition of (possibly unavailable) services. Indeed, availability (or the lack thereof) can genuinely change the behavior of a system. Nevertheless, many Web service applications lack a directory or fail to use UDDI: the Web services standards simply serve as a protocol between nodes in a loosely coupled distributed application.

A. Issues Related To Requirements Specification for Service based Systems

The following six sections present challenges that appeared in the case organizations. These challenges might turn out to be obstacles to the successful implementation of the RE processes if organizations do not pay special attention to them.

1) Duration of the change

All the case organizations showed that the introduction of RE and the organization-wide implementation of the RE process take time. In particular, cultural change took place more slowly than the practitioners were prepared for. Product development personnel were so used to focusing on technical details that they had difficulties defining user requirements. The change was deeper and more time- consuming if people had beliefs and attitudes related to the systematic definition of user requirements.

2) Personnel changes

Personnel changes took place in all the case organizations during the RE process improvement. For example, in one case organization, a representative of the senior management was actively involved in the

project of the RE process improvement. Because of the organizational change, this manager could not continue to support the RE
Process improvement. In this same case organization, the manager of the process improvement project changed.

3) Training and support resources

'Just-in-Time' training combined with RE support was piloted in at least two product development projects in each case organization. The experiences showed that 'Just-in-Time' training and RE support require persons that have good knowledge about the RE process and practices of the organization. In addition, experience of product development and knowledge about application domain increased these persons' capability to help project teams to tailor the RE process and to apply RE practices to the needs of the projects.

4) Scope of the RE process

All the case organizations integrated the RE process with their product development process. The experiences showed that RE relates also closely to business processes as strategic planning and road mapping. In addition, the experiences showed that sales, sales support, and marketing personnel have valuable knowledge and information for customer and user requirements definition. However, it was occasionally difficult to involve these important stakeholders in requirements definition because they could not see how to combine the RE practices with their existing duties and tasks.

5) Implementation of a RE tool

All the case organizations were interested in acquiring a RE tool. One case organization defined the RE process first, and subsequently acquired a RE tool. The use of the tool made a set of RE practices systematic and supported the organization-wide implementation of the RE process. Some practitioners were very satisfied with the tool, while others would have improved its implementation. The practitioners emphasized that their own company had to invest in tailoring the RE tool to the needs of product development projects. They wanted to have, for example, company specific document templates and reports ready for product development projects.

6) Measurement

None of the case organizations was able to set measurable goals for RE process improvement, neither were they able to link process improvement goals with business objectives. The main reason for the difficulties in setting measurable goals was that the organizations did not have quantitative data from their existing RE practices. On the other hand, it is unrealistic to expect organizations that are just starting to improve their RE process to have measured their RE practices.

III. PROPOSED WORK

In proposed work we have been using the objectiver goal driven approach to easier access of information in web services. The approaches to obtain quality are to identify all the goals pursued by the project stakeholders and all involved people, link them to higher-level strategic goals and define how goals can be achieved. The use of a highly-graphic notation facilitates are talking about project requirements and goals, quick identification of concept interrelationships and validation of the model.

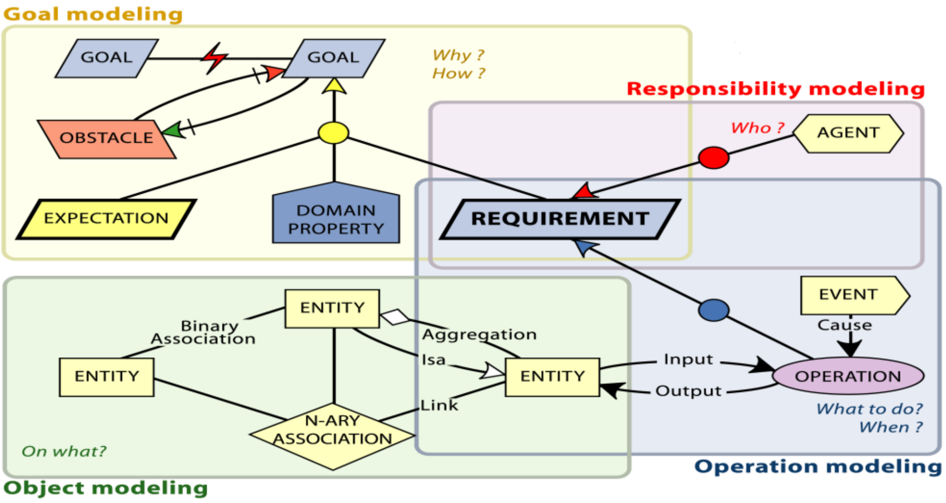


Figure 2. Objectiver goal driven approach

A. Goal Model

Goals are objectives which a system should achieve through cooperation of actors in the intended software and in the environment; Goal modeling is especially useful in the early phases of a project.

The main objective is to achieve the quality while accessing the information in web applications. There are several reasons to fails the projects are Lack of user input, Lack of executive support technology, Unrealistic expectations, and unclear objectives. Therefore the failure can be overcome by the objectiver goal-driven approach to requirements engineering.

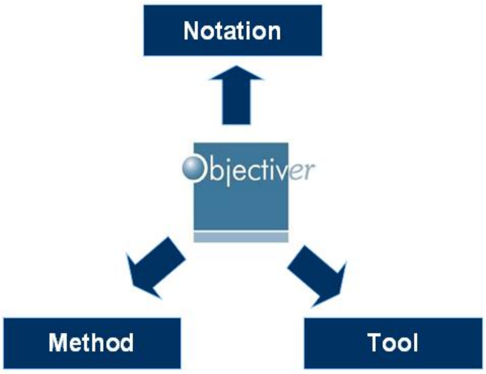


Figure 3. Objectiver goal driven approach process

The objective is to studying the problem and its environment rather than by specifying the solution straight away and links the project requirements to the business strategy so easier to establish traceability from business processes to business goals. This results in finding out how changes to business goals shall impact your processes and project requirements. Projects may consider how the intended system meets organizational goals, why the system is needed and how the stakeholders' interests may be addressed.

A goal model:

- Expresses the relationships between a system and its environment (i.e. not only on what the system is supposed to do, but why). The understanding this gives, of the reasons why a system is needed, in its context, is useful because "systems are increasingly used to fundamentally change business processes rather than to automate long-established practices".
- Clarifies requirements: Specifying goals leads to asking "why", "how" and "how else". Stakeholders' requirements are often revealed in this process, with less risk of either missing requirements, or of over-specifying (asking for things that are not needed).
- Allows large goals to be analyzed into small, realizable goals:
- Deals with conflicts: goal modeling can identify and help to resolve tradeoffs between cost, performance, flexibility, security and other goals. Goal modeling can also reveal divergent interests between stakeholders. Goal modeling can identify conflicts because meeting one goal can interfere with meeting other goals.
- Enables requirement completeness to be measured: Requirements can be considered complete if they fulfill all the goals in the goal model.
- Connects requirements to design: for example, the i* "Non-Functional Requirements (NFR) framework" uses goals to guide the design process.

1) Goal oriented approaches in RE

Despite the fact that there is no common definition of the RE process, four tasks to be performed have been identified.

- requirements elicitation
- requirements negotiation
- requirements specification, and
- requirements validation

Requirements elicitation is about understanding the organizational situation that the system under consideration aims to improve and describing the needs and constraints concerning the system under development. The relevant knowledge about the problem (system) is typically distributed among many stakeholders. The objective of negotiation is to establish an agreement on the requirements of the system among the various stakeholders involved in the process. Requirements specification involves a mapping of real-world needs onto a requirements model. The wider view on RE suggests that requirements specification goes beyond the traditional approach which concentrates on functional requirements modeling and involves the modeling of the enterprise context which provides the purpose of the intended system. Finally, the validation task intends to ensure that the derived specification corresponds to the original stakeholder needs and conforms to the internal and/or external constraints set by the enterprise and its environment. Examination of current research has highlighted a large number of cases where goal analysis techniques have proven successful in different RE contexts.

The Goal model describes below describes for Online Banking services in order to access the account and also defines the high level goals to improve the performance. The performance has been increased by providing security to web services. Thus login password has been requested and that has been encrypted for security to access the services. And also we are also providing privacy for preserving account to access the account by special encryption technique 128-bit SSL for transmission.

The expectation is to be customer is to keep the password secret. When alternatives are identified, further interviews and analysis shall be conducted to decide which solution shall be preferred over the other one.

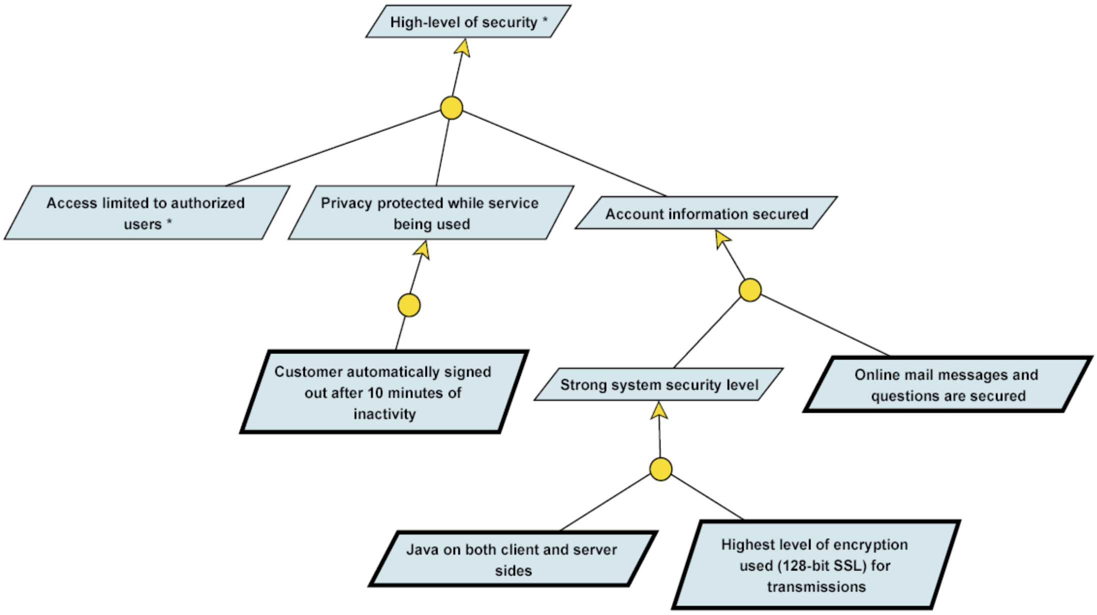

High-level of security *
Access limited to authorized users *
Privacy protected while service being used
Account information secured
Customer automatically signed out after 10 minutes of inactivity
Strong system security level
Online mail messages and questions are secured
Java on both client and server sides
Highest level of encryption used (128-bit SSL) for transmissions

Figure 4. System design for goal model

B. Responsibility Model

Objective here is to identify agents and assign them responsibility for the expectations and requirements identified in the goal model. The actors involved in responsibility model are

Agent: a human, device or system component

- System agent: part of the system being modeled
- Environmental agent: part of the system environment

Requirement: a low-level goal placed under the responsibility of a unique system agent

Expectation: assigned to an environmental agent. As such expectations cannot be enforced by the system.

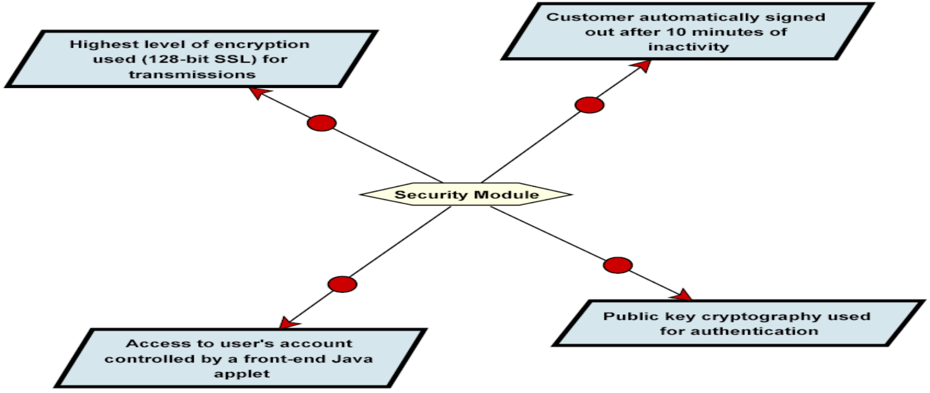

Highest level of encryption used (128-bit SSL) for transmissions
Customer automatically signed out after 10 minutes of inactivity
Security Module
Access to user's account controlled by a front-end Java applet
Public key cryptography used for authentication

Figure 5. System design model for requirement engineering

Some goals may conflict with each other under certain circumstances.The solutions to overcome these conflicts: Introduce new goals/requirements that shall prevent the conditions leading to the conflict from occurring Opt for an alternate solution (if we can't live with this conflict in the system). The obstacle is to identify conflicts that prevent reaching any goal, and new goal to resolve or alleviate the obstacles. Thus the obstacles can be identified by refining and resolving the specific obstacle separately. The encryption has been used to encrypt the data that has been transmitted.

C. Object model

The object-oriented model is based on a collection of objects. An object contains values stored in instance variables within the object. Thus objects contain objects to an arbitrarily deep level of nesting. *Attributes/properties:* objects will have at least one attribute. Possible slot types are primitive types (integer, Boolean, string etc.), references to other objects (modeling relationships) and sets of values of these types. An object also contains bodies of code that operate on the object. These bodies of code are called methods. *Method/Operations:* They are attached to classes or slots and contain Meta information, such as comments, constraints and default values. *Relationship/relations:* they represent the relation between objects/classes from object model (KB). Major classes of relations exist: relations combining labels(the name we tend to give to things) and concepts(the things themselves) and concepts and relations combining concepts (the part-whole relation). Objects that contain the same types of values and the same methods are grouped into *classes*. A class may be viewed as a type definition for objects. Analogy: the programming language concept of an abstract data type. The only way in which one object can access the data of another object is by invoking the method of that other object. This is called sending a message to the object. Internal parts of the object, the instance variables and method code, are not visible externally or some researchers called it as black box. The following Fig shows a simple Banking System object Model, containing classes for Head-Office, Branch, Accounts held at that Branch, and the Customers who the Accounts belong to. Object/Class represents the tangible things. For example, an object representing a *bank account*. The object contains instance variables *number* and *balance*. The object contains a method *pay-interest* which adds interest to the balance. Under most data models, changing the interest rate entails changing code in application programs. In the object-oriented model, this only entails a change within the *pay interest* method. In commonly-known object-oriented data models attributes and associations are not defined with the class specification itself [14]. Instead, class properties are first-class primitive themselves.

Many claims have been made about the object-oriented model These claims include:

- ease of understanding object-oriented models due to a consistent underlying representation throughout the development
- process
- the ability to model the behavior of objects (encapsulation of data and process)
- ease of modification and extensibility of object-oriented models.
- ease of reuse of object components from previously designed systems
- superior data abstraction facilities including inheritance and polymorphism

Thus the model domain includes entities and agents. The Fig.5 explains the object model for banking model. This model has specialization, associations, aggregation, etc., to represent entities.

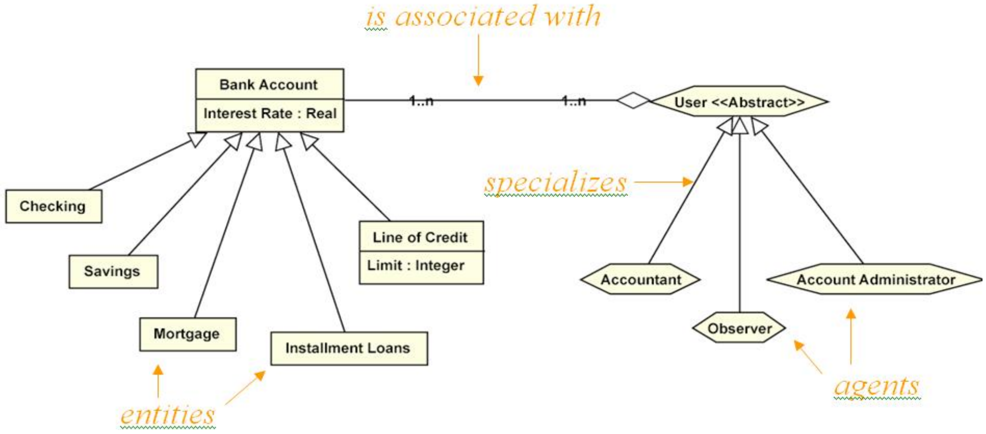


Figure 6. Object design model for requirement engineering

D. Operation model

An operating model can be used as a framework for formulating operations strategy – explicit choices about the best deployment of organization's elements to achieve the business goals. It is usually informed by the

business model. In some cases, an operating model can become the source of competitive advantage and can inform the business model.
The operating model can be a vehicle to describe how the organization does business today - the "as-is" operating model. This can be the foundation for an organization that wants to transform its business. New business drivers can be translated into requirements that can lead to a new "to-be" or "target" operating model. The target operating model represents the high level requirements that drive the future business and IT architecture design.

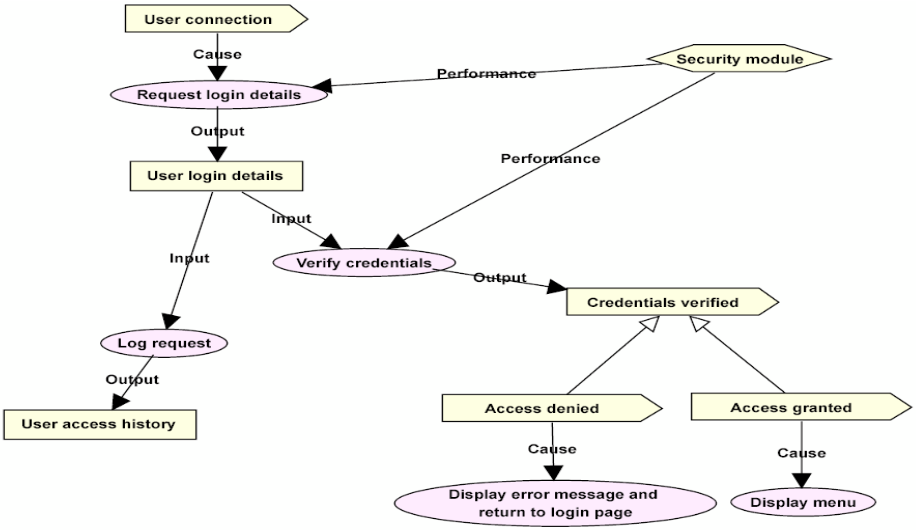
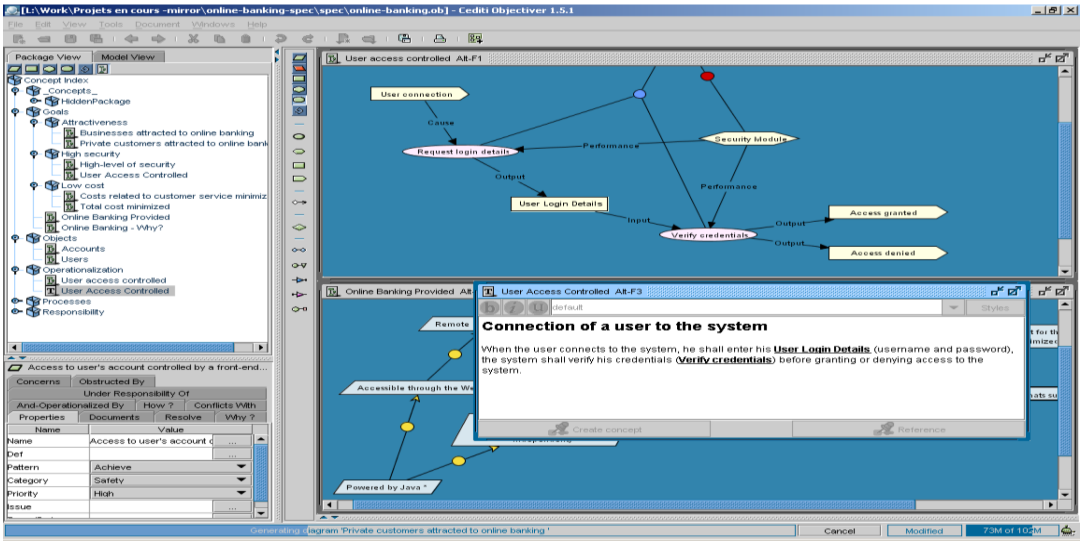


Figure 7. Object design model for requirement engineering

IV. EXPERIMENTAL ANALYSIS

The objectiver tool which has been implemented to calling the tender to managed easily and provides the traceability to achieve business goals. Thus provide the highly integrated views on the model and provide powerful querying tool for validation, verification and correctness. Thus the implementation of objectiver tool which has been used for the corrections of the user to achieve the quality of service.
The alternatives and expectations have to be made to correctness of the services to achieve the access of services. The Fig. 8 and Fig. 9 is the implementation and web generated document for the connection of user to system.



Concept Index
Concepts
HiddenPackage
Goals
Attractiveness
Businesses attracted to online banking
Private customers attracted to online bank
High security
High-level of security
User Access Controlled
Low cost
Costs related to customer service minimiz
Total cost minimized
Online Banking Provided
Online Banking - Why?
Objects
Accounts
Users
Operationalization
User access controlled
User Access Controlled
Processes
Responsibility

Access to user's account controlled by a front-end...

Name	Value
Name	Access to user's account c
Def	
Pattern	Achieve
Category	Safety
Priority	High
Issue	

User Access Controlled Alt-F3

default

Connection of a user to the system

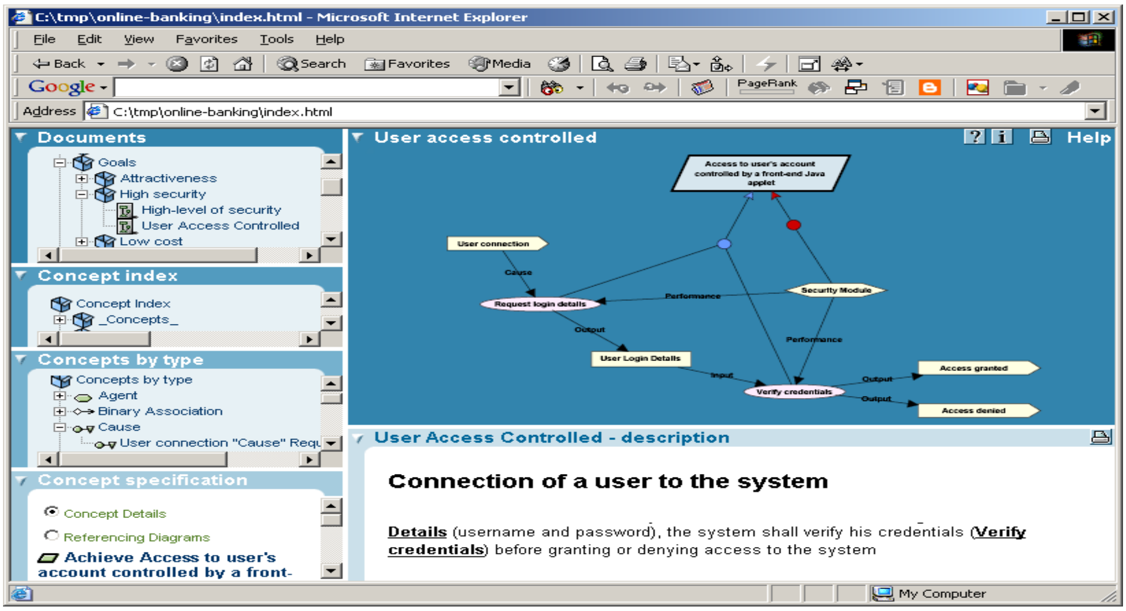
When the user connects to the system, he shall enter his **User Login Details** (username and password), the system shall verify his credentials (**Verify credentials**) before granting or denying access to the system.

Generating diagram 'Private customers attracted to online banking '

Figure 8. Tool implementation

A. Web generated document

The document has been generated automatically and has been allowed the authorized user to access the web page. The username and password has been used to access the page. And the system itself verifies the attributes and check assessment of authorized user before accessing or denying access to the system.



Connection of a user to the system

Details (username and password), the system shall verify his credentials (**Verify credentials**) before granting or denying access to the system

Figure 9. Generated Document

V. CONCLUSION

In this paper we have introduced the objectiver approach tool to increases the quality and security service based systems. Thus the requirement engineering has been used to provide the verification, validation and correctness Most specially we have proposed the online banking services in order to provide services to access the transactions. As in future work we intended to introduce another tool to increases the traceability and verifiability.

REFERENCES

[1] Binde Luo, Zhiyong Shi, Dawei Ma, YIN Shi-ping. Design and Capacity Analysis of PoC Service based on TD-SCDMA Network, The 6th International Conference on Wireless Communications, etworking and Mobile Computing[C], 2010.

[2] Vehmas A. "Service adoption for push over cellular"[EB/OL], 2004, web page: www.tml.hut.fi/Opinnot/T-109.551/2004/reports/poc.pdf

[3] D. Roman, J. Scicluna, C. Feier, M. Stollberg & D. Fensel, "Ontology-based Choreography and Orchestration of WSMO Services", http://www.wsmo.org/TR/d14/v0.1/, (online accessed= Jan 2011)

[4] Kevin Carey, Vincent Wade," Realizing Adaptive Web Services through Automated Policy Refinement", IEEE, 2007

[5] Qiao Xiaoqiang, Wei Jun, "A Decentralized Services Choreography Approach for Business Collaboration, IEEE International Conference on Services Computing (SCC'06), 2006.

[6] M.P Papazoglou , P. Traverso, Schabram Dustdar & F. Leymann, (2007) ," Service-Oriented Computing: State of the Art and Research Challenges", Journal of Innovative Technology for Computer Professionals, IEEE Computer Society, November 2007.

[7] Erradi A., "Policy-Driven Framework for Manageable And Adaptive Service-Oriented Processes", PhD thesis, The school of Computer Science and Engineering(CSE), University of New South Wales(UNSW), June 2008.

[8] Martin, D., et al.,(Nov 2004), OWL-S: Semantic Markup for Web Services. DAML White Paper Release.

[9] Pahl,"Dynamic Adaptive Service Architecture .Towards Coordinated Service Composition", Lecture Notes in Computer Science, Software Architecture. M. Babar I.Gorton, Springer Berlin / Heidelberg. 6285:472-475, 2010, [doi: 10.1007/978-3-642-15114-9_43]

[10] Souttar, J. Three Clicks and it's gone. Eye, The international review of graphical design, No. 36, Vol. 9, summer 2000, p.86

Theses

[11] K.Kritikos, **"**QoS-Based Web Service Description and Discovery," PhD thesis, Computer Science Dept., Univ. of Crete, Dec. 2008.

Proceedings Papers

[12] Mohammad Alrifai and Thomas Risse, "Combining global optimization with local selection for efficient QoS-aware service," In *Proc. WWW'09 Proceedings of the 18th international conference on World wide web*, Published by ACM 2009 Article.

[13] L. Taher, H. El Khatib, R. Basha, "A framework and QoS Matchmaking Algorithm for Dynamic Web Services Selection (2005)," In *Proceedings of the 2 nd International conference on Innovations in information Technology(IIT'05).*

[14] Dongyun Liu and Hong Mei, "Mapping Requirements to Software Architecture by Feature-Orientation," in *Workshop on software Requirements to Architectures(STRAW)*, pp - 69-76, ICSE 2003 : Portland, Oregon, USA.

[15] David and S. Masoud, "Mapping Non-functional Requirements to Cloud applications," In *Proceedings of the 2011 International Conference on Software Engineering and Knowledge Engineering (SEKE 2011)*, Miami, Florida. *(Accepted for publication).*