REVUE ALGERIENNE DES SCIENCES -A
Sciences de la Nature et de la Vie
Sciences Techniques
ISSN ; 2661-7064 http://umiv-eltarf.dz/fr/



Toward a Combined Approach for Requirements Engineering of SaaS

Farida Kherissi

Université Badji Mokhtar , Faculté de Sciences de I'lngéniorat, Laboratoire LISCO, B.P N ^{0} 12, Annaba, Algérie, 23000.

Informations

Mots clés :
SaaS
VORD

Combining

Requirements

*Correspondance :
farida.kherissi@univ-annaba,dz

Regu le jj/mm/aa,
Révisé jj/mm/aa,
Accepté lei)/mm/aa

Résumé

The Software as a service (or SaaS) is a way of delivering applications over the Internet-as a service. Despite its increasing acceptance within the industry, many important questions remain unanswered. Issues related to the
best software architectures decisions for cloud-based systems are faced with the question of appropriate techniques applying at early phase like requirements engineering. Requirements Engineering (RE) is one of the core Software Engineering activities. The Requirements elicitation is very important and non-trivial task in the Requirements Engineering activities. The paper highlights problems in Service oriented Requirements Engineering (SoRE), and presents guidelines toward a hybrid approach for elicitng requirements based on VORD viepoints RE method and i* model of the GoRE.

1. Introduction

Cloud computing is an emerging paradigm in which information technolow resources are provided over the internet as a service to users. Software-as-aService (SaaS) is offered by cloud, which can be delivered in a composite form, consisting of a set of application and data components, that works together to deliver higher-level functional software. Software as a Service (SaaS) is a cloud computing service model in which applications are delivered to customers as a service. Instead of developing and maintaining a version of application code for each individual tenant, SaaS application serves thousands of tenants with one single application instance [35]. The main principle is offering computing, storage and software as a service. SOA (Service Oriented Architecture) is an architectural paradigm that has recently become one of the preferred choices for designing and developing systems that are characterized by coarse-grained services and service consumers [351. One of the key motivating factors behind the adoption of this paradigm is the capacity to deliver flexible ICT (information and communication technologies) solutions, which have the ability to respond quickly and prove cost-effective to changing business or consumer requirements.

Service oriented software engineering (SOSE) advocates software reuse while aiming to achieve better alignment of software solutions to business requirements. Service orientation has evolved from Object Oriented Analysis and Design (OOAD) and Component Based Software Development (CBSD), the major difference being that reusable artefacts are in the form of services rather than objects or packaged components [361.

Requirements Engineering (RE) is accepted as one of the most crucial stages in software design and development as it addresses the critical problem of designing the right software for the customer. Traditionally RE was considered to be a sequential process that starts with requirements elicitation, and proceeds to modeling, analysis, specification, agreement, along with evolution and management [14].

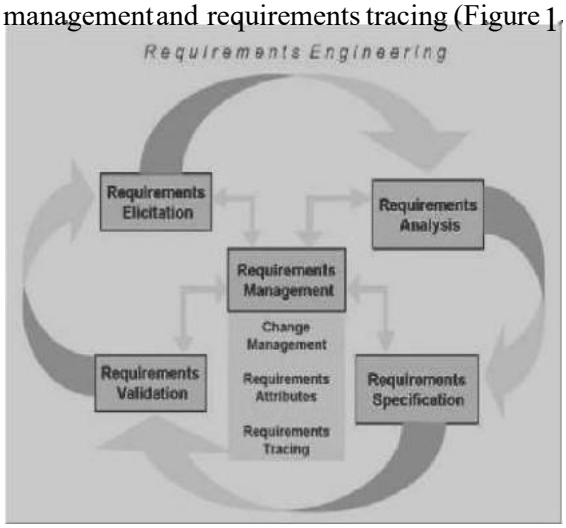
In SOA, Service oriented Requirements Engineering (So RE) shares some activities with traditional Requirements Engineering but its focus is on the identification of services and workflows used to modeling applications (developed in a SOA framework and running in a SOA framework) and on their reuse. SoRE is different from traditional RE process in various aspects. The difference between traditional and SoRE activities is essentially in performing the service and workflow discovery [37], identifying the service requirements specified in service level agreements (SLA), identifying and specifying the dynamic requirements that may vary at runtime.

The service-oriented systems are analogous to a federation; i. e., a system consists of several constituents, which remain independent in internal and sometimes even in external matters. The constituents are under control of different stakeholders, which can have quite different goals. There are at least three paradigm related roles: service consumer, service infrastructure owner, and service owner. The goals of service consumer are aligned with the business goals. The goals of infrastructure owners are related to the development and maintenance of a service system infrastructure. The goals of service owner concern the realisation of capabilities [37].

Cloud computing brings numerous challenges in this area since the traditional methods need to be adapted and new RE methods [37], including the requirements modelling, has to be investigated. More specifically, the success of adopting the new paradigm highly depends on the degree to which requirements are correctly understood by both service providers and consumers. Conventional RE approaches are not always well adapted to the service-oriented paradigm where the consumer is not always available for the traditional interview and observation-based requirements elicitation process or even the consumer does not have any requirements yet and will discover them once using the service. The service provider has to create requirements in order to attract consumers and to compete in the market. Besides, even if the requirements or wishes exist, very often they are fuzzy and volatile and the service provider has to take into consideration their evolution and management process. Therefore, the need for approaches enabling the elicitation of service consumers wishes, needs and requirements becomes more and more pressing [34], [35], [36], [37]. We think that the elicitation process is a challenging task in SORE, because of both : a very few works at early RE phase are proposed ans its crucial importance in the development process.

It is widely accepted that a multi-perspective approach to requirements engineering can, potentially, lead to requirements specifications which are more likely to satisfy the needs of a diverse set of system stakeholders [2], [3], [8], [37]. The viewpoints approach recognizes that all requirements cannot be discovered by considering system from a single perspective; it is essential to collect requirements from different classes of

system end-users and other stakeholders, to management and requirements tracing (Figure 1.



collect information of different types a bout the application domain, system environment and system development. The viewpoint based approach to requirements engineering has been approved in traditional software systems development context. A viewpoint is an encapsulation of partial information about a system's requirements. Information from different viewpoints must be integrated to form the final system specification [2]. Two different kinds of viewpoi_nt have been proposed: Viewpoints associated with system stakeholders, and Viewpoints associated with organisational and domain knowledge [21. PREeview viewpoints are flexible, generic entities which can be used in different ways and in different application domains. VORD [43] is a pragmatic adaptation of the use of viewpoints for requirements engineering that does not propose any particular types of viewpoint or notation.

2. Requirements Engineering

The term "Requirements Engineering" appears to have been used for the first time By Hagelstein 111 [1]. All projects begin with a statement of requirements. Requirements are descriptions of how a software product should perform. A requirement typically refers to some aspect of a new or enhanced product or service [13].

The requirements engineering process involves a clear understanding of the requirements of the intended system. This includes the services required of the system, the system users, its environment and associated constraints. This process involves the capture, analysis and resolution of many ideas, perspectives and relationships at varying levels of detail [8].

Requirements engineering refers to all life-cycle activities related to requirements. This primarily includes gathering, documenting and managing requirements (Figure 1.). With the growing awareness of the significance of requirements in the software process, requirements engineering increasingly becomes an area of focus in software engineering research. Common requirements engineering activities are elicitation, interpretation and structuring (analysis and documentation), negotiation, verification and validation, change

Requ1rem9nrs Engineering

Figure 1. Requirements Engineering Activities

The literature offers several approaches to requirements engineering. Each of them focuses on specific activities of the requirements engineering process, as :

- The scenario-based approach [11]
- The GORE (Goal Oriented Requirements Engineering) [191
- Aspect-oriented requirements engineering approach [16]
- Problem Frames approach [10] .
- ...etc.

2.1 Requirements Elicitation

Quality of the software are contingent to requirement elicitation, requirement analysis and requirement management [42] Defining requirements for software is a process that involves a wide range of human, technical and methodological skills, combining rigor and creativity. An effective requirements study significantly reduces the cost of development and maintenance of an application and increases its quality [12], [13].

Requirements elicitation is the process of seeking, uncovering, acquiring, and elaborating requirements for computer based systems. It is generally understood that requirements are elicited rather than just captured or collected. This implies there are discovery, emergence, and development elements to the elicitation process. Requirements elicitation focuses on the collection of requirements from different stakeholders. Collecting software requirements activity is considered by various levels of users as difficult, time consuming, and inefficient. Abilities of communication services within the intended organization going to conduct software requirements elicitation has a major role to play in this context and it affects considerably consequent activities, tasks, and processes [31]. People's opinions differ completely and significantly during this activity due to naturally existing flavors between individuals. Knowledge for systems must be derived from expert sources and elicitation of right knowledge can take several years and cost

huge amount of money [42] especially in case of expert systems or knowledge-based systems. The software requirements process, represents one of the most critical phases of the software development lifecycle [31]. The main objective of the elicitation techniques is to support the maximum level of stakeholders' collaboration, cooperation, and contribution. We give the most important elicitation techniques cited in [31]: RESCUE creativity techniques that are humanintensive requirements generation.

-Search-based technique: an approach intends to reduce RE problems, like requirements selection and prioritization, to search problems, and then apply multi-objective search-based techniques to lead the search of an optimum.

-POSE: a quality-oriented tool for requirements elicitation. Its function is to facilitate the elicitation of quality requirements through interactive simulation of the system which is going to-be maintained or developed.

-Spearmint: provides capabilities for classifying and representing process models in an application environment that is web based where its power rests in the Electronic Process Guide generator (EPG) tool to create process specifications and it provides direct support for the process engineer. Spearmint directly impacts efforts for documentation of processes in process improvement.

-Joint Application Development CAD): brings users and technical professionals together and is designed mainly to discover end user's functional requirements (FR) in software development. JAD gathers a broad range of stakeholders together in highly structured and focused meetings to gather information and specify requirements. The focus of JAD is a structured workshop known as the JAD Session, in which all stakeholders design a system.

2.2 Requirements Engineering Based Viewpoints

22.1 Viewpoint Approach

A viewpoint can be informally defined as an angle from which a domain can be observed [31]. The use of viewpoints in the requirements engineering activities is well known [29], [18]. This approach recognizes that all requirements cannot be discovered by considering system from a single perspective; it is essential to collect requirements from different classes of system end-users and other stakeholders, to collect information of different types — about the application domain, system environment and system development. There are a number of different models of viewpoints. We consider a viewpoint to be a role performed by a stakeholder when examining a system. The viewpoint based approach to requirements engineering has been approved in traditional software systems development context [18]. The goals of service owner concern the realization of capabilities. In the viewpoint-based methods which have been developed, two different kinds of viewpoint have been proposed:

Viewpoints associated with system stakeholders. A system stakeholder is anyone who is, directly or indirectly, affected by the existence of a system (end-users of a system, managers of 410rganization, other human and computer-based systems in an organization external entities who have some kind of interest in the system and engineers involved in the design, development and maintenance of the system.

1. Viewpoints associated with 420rganizational and domain knowledge which constrains the system requirements. The constraints may be physical (e.g., network performance), 420rganizational (e.g., incompatible hardware used in different divisions of a company), human (e.g., average operator error rate) or may reflect local, national or international laws, regulations and standards. This type of viewpoint cannot be associated with a single class of stakeholder but includes information collected from many different sources (people, documents, other systems, etc.). The viewpoints have many advantages:

- They explicitly recognize of the diversity of the sources of requirements
- They provide a mechanism for organization and structuring diverse information
- They provide a means for requirements sources or stakeholders to identify and check their contribution to the requirements.

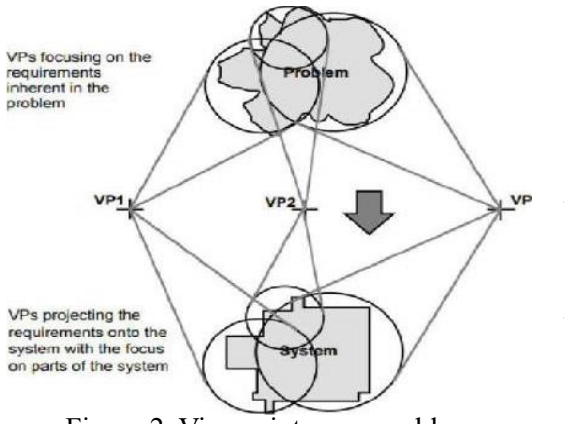


Figure 2. Viewpoints on a problem

2.2.2 VORD Method

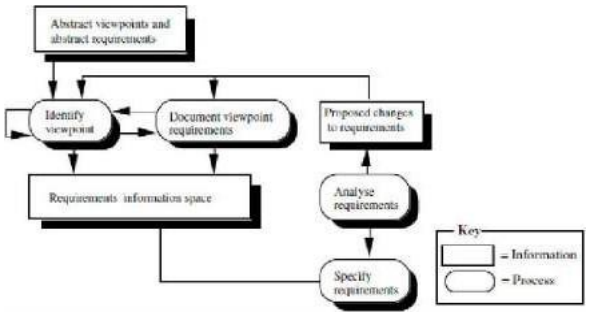


Figure 3. The Vord Process Model

The viewpoint VORD encapsulates a set of attributes that help define and structure specifications. Briefly, a viewpoint template includes the following components: Identifier, Label, Type, Attributes, Requirements, Event scenarios (Figure 4.).. The VORD method is useful in the detection of user needs, and also in identifying services that the user expects the system.

Figure 4. Viewpoints Notation

VORD viewpoints fall into two classes:

Direct viewpoints. These correspond directly to clients in that they receive services from the system and send control information and data to the system. They are either system operators/users or other systems which are interfaced to the system being developed.

Indirect viewpoints. Indirect viewpoints have an 'interest' in some or all the services which are delivered by the system but do not interact directly with it. They may generate requirements which constrain the services delivered to direct viewpoints.

2.3. The Goal Oriented Requirements Engineering

2.3.1 Goal Oriented Requirements Engineering Goal-Oriented Requirement Engineering (GORE) is a subarea of Requirement Engineering (RE), which addresses using of goals for eliciting, elaborating, structuring, specifying, analyzing, negotiating, documenting, and modifying requirements [2 3].Goal-oriented req uirements engineering (GORE) [19], [23] is concerned With the use of goals. Goals express intention and capture the reason of the system to be built. In general, each high-level goal can be considered to belong to one of the following comprehensive categories: enterprise goals (reflecting general policy,

Generalisarion

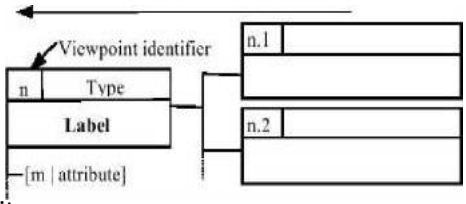
strategy and tactics), problem-solving goals and innovative goals [19]. Goals also vary in type, ranging from representations of functional properties the system must offer to non-functional concerns related to quality of service [19].

2.3.2. The 1* Method

The i* [19] approach provides a description of work organisation in terms of dependency relationships among actors. This approach acknowledges the fact that actors have freedom of action, within the social (inter-actor) constraints, called strategic dependencies.

CD

Figure 5. Example of a Goal Model Illustrating its



Basic Concepts

3. Requirements Engineering in SOA Systems

The early SORE approaches were derived from the initial phases of traditional software development methodologies, and the later ones are original, taking into account the specific characteristics of service-oriented systems [37].

A systematic literature review process identified a result that shows that most publications have been published in conferences (86%), follows to journals (14%). Despite the count of selected studies is not statistically significant, the authors suspected that most part of publication is in conferences because the approaches are not mature enough, maybe because the research area is still recent [33].

The proposed in recent years can be partitioned into two groups: the first group refer to slightly revised traditional approaches, and the second one includes innovative approaches [3, 4, 15, 16, 17, 22, 23, 26, 28, 29, 31, 32,331.

While the offer of RE approaches for software and information systems development is quite rich, the perspective of service-oriented development and service engineering is rather neglected, especially from the service provider's point of view [23].In

[21] a method to analyze requirements is proposed.

It can inform users of potential errors in requirements by detecting the correctness of requirements, which is driven by users' personalized operations on goal models, and customize personalized processes to satisfy users' requirements by reusing domain processes. The personalized processes are the basis for Web service discovery and composition. J. Ralyté introduce in [231 a .framework to support RE for service engmeerlng.

In [181, authors focused on RF phase of SOSE and proposed a SORE framework. Xiang et al. have proposed Service Requirement Elicitation Mechanism (SREM), which is based on Service Requirements Modeling Ontology (SRMO), which has taken basic concepts from agent-oriented modeling framework i* (i star) [42]. Value based software development concepts are used in Value Gap Model for eliciting requirements for service components [42].

4. The Avantages of Using a Combining of VORD and I*

Where viewpoints are used to support requirements elicitation, they are clearly outside of the system being specified and are, primarily, the sources of the system requirements. At this stage, only informal descriptions and models of the system are likely to be available [43].The GoRE (Goal oriented Requirements Engineering) methods provide a range of techniques for goal identification, offers a set of heuristics [44] for elicitation of requirements through the decomposition of goals, enabling alternatives...etc; on the other hand, VORD is an approach leading to viewpoint-oriented requirements engineering geared specifically to integration, lacking a more systematic mechanism to direct the process of elicitation requirements. Although, we think that a hybrid approach using i* to model viewpoints extracted from stakeholders is an important issue in SoRE for many reasons:

1. The VORD method is a subjective approach and do not use a graphical model, so we can complete the requirement analyze VORD with a mapping to a goal diagram.

2. According to Sommerville and Sawyer in [6], some of the disadvantages of using viewpoints are inflexible viewpoint models, fixed notations for requirements, limited support for requirements evolution, and limited support for requirements negotiation.

The evolution/change of elicited requirements can easily modelled using alternatives with ANI)/OR links in a goal 1. diagram, that's not possible in viewpoints approaches.

2. The most important motivation is trying to gain advantages of both VORD and i* in one hand, and dealing with a complete approach for requirements engineering SaaS.

5. Conclusion

Service-oriented systems engineering in-the-small can be considered as an integral part software engineering.with the questlon of appropriate techniques applying at early phase like req uirements engineering.

RE for Service Oriented Software Development can involve activities of traditional process like modeling, specification, and analysis but the way in Which these activities are carried out is different.

The Service oriented Requirements Engineering (SoRE) is more cornplex than traditional RE, having new specific problems, as human involvement in software depelopment, problems of alignment of requirements and services, ...etc. Although, Requirements elicitation involves activities that are intensely communicative.

To our best knowledge, there is no a systematic approach or work for eliciting activities in SoRE . In this paper, we present a review of two traditional RE approaches : Viewpoints and We think a of combining of them is an important issue for SoRE.

References

Hagelstein J.(1988) Declarative Approach to Information Systems Requirements,in KnowledgeBased Systems, 4.

[2] G. Kotonya and l.Sommerville.(1992) Viewpoints for requirements definition, .1. Snftware Eng.,7: 375-387.

[31 A. Finkelstein. J. Kramer, B. Nuseibeh, L. Finkelstein, and M. Goedicke(1992) Viewpoints: A framework for integrating multiple perspectives in system development, fnt J. Snffnwe Eng. Knowl. Eng, 2:31-58.

Nuseibeh, B., J. Kramer, and A. Finkelstein(1994) A Framework for Expressing the Relationships Betw•een Multiple Views in Requirements Specification, Transaction on Software Engineering, 20(10): 760-773.

15] Gerald Kotonya and [an Sommeiville(1996) Requirements engineering with viewpoints, Software Engineering Journal

Ian Sommerville and Pete Sawyer(1997), Viewpoints: principles, problems and a practical approach to requirements engineering, Annals of Software Engineering 3

[71 Colette Rolland, Carine Souveyet et Camille Ben Achour, Guiding Goal Modeling using Scenarios. IEEE Trans. Softw. Eng., vol. 24, pages 1055-1071, December 1998.

[81 I. Sommerville, P. Sawyer, and S. Viller(1998) Viewpoints for requirements elicitation: a practical approach, 74-81.

[91 ROLLAND C, PRAKASH N and BENJAMEN A (1999) A Multi-Model View of Process Modeliing, Requirements Engineering Journal, 169-187.

[10] A. Cockburn (2001) Writing Effective Use Cases, Addison Wesley, New York.

[Ill Michael Jackson (2001) Problem frames analyzing and structuring software development problems. Addison-Wesley Longman Publishing Co.,lnc., Boston, MA, USA.

[12] Alistair Sutcliffe (2003) Scenario-Based Requirements Engineering. In Proceedings of the 1 Ith IEEE International Conference on Requirements Engineering, Washington, DC, USA,IEEE Computer Society.

[131 Lau D., Mylopoulos J .(2004) Designing web services with Tropos. In: Proceedings of the IEEE International Conference on Web Services

[ICWS'041, 6-9 June, san Diego, California, USA, IEEE Computer Society,. 306-315.

[14] A. Aurum and C. Wohlin, editors (2005) Engineering and Managing Software Requirements. Springer Verlag, Berlin Germany,.

[15] Loucopoulos, P.(2005) Engineering and Managing Software Requirements. Springer, Heidelberg.

[16] Papazoglou M. P„ van den Heuvel W.-l.(2006) Service-oriented design and development methodology. International Journal of Web Engineering and Technology, 2(4) :412-442.

[17] Decker, Bjorn, et al. (2007), Wiki-Based Stakeholder Participation in Requirements Engineering„ IEEE SOFTWARE , Fraunhofer Institute for Experimental Software Engineering.

[18] Aais Rashid et Ruzanna Chitchyan.(2008) Aspect-oriented requirements engineering roadmap. In Proceedings of the 13th international workshop on Early Aspects, EA '08, pages 35—41, New York, NY, USA. ACM.

[19] Report Working Conference on Requirements Engineering. Paech, Barbara, Heymans, Patrick and Persson, Anne. Montpellier : ACM SIGSOFT Software Engineering Notes, 2008 . Requirements Engineering Foundation for Software Quality (REFSQ'08). pp. Volume 33, Number 5, Page 2226.

120] J. Wang, K. Q. He, P. Gong, C.Wang, R. Peng and B. Li (2008) RGPS: A Llnified Requirements MetaModeling Frame for Networked Software, Proceedings of Third International Workshop on Advances and Applications of Problem Frames (IWAAPF 2008) at 30th International Conference on Software Engineering (ICSE 2008), pp. 29-35.

[211 Crabtree, Carlton, Seaman, Carolyn and Norcio, Anthony. (2009) Exploring Language in Software Process Elicitation: A Grounded Theory Approach. : IEEE Computer Society,. Proceedings of the 2009 Third International Symposium on Empirical Software Engineering and Measurement (ESEM '09).

[22] Sommerville, [an.(2009) Software Engineering. s.l. Pearson.

[23] Axel van Lamsweerde(2009) Requirements Engineering : From System Goals to UML Models to Software Specifications, Wiley.

[24] Lewis, Grace; Smith, Dennis; & Kontogiannis, Kostas.(2010)A Research Agenda for ServiceOriented Architecture (SOA): Maintenance and Evolution of Service-Oriented Systems. CMU/SEI2010-TN-003. Software Engineering Institute, Carnegie Mellon University.

[25] H. Chen and K. He(2011) A method for serviceoriented personalized requirements analysis, Journal of Software Engineering and Applications,

[261 Verlaine B., Jureta 1., Faulkner S.C 2011) Towards conce ptual foundations Of requirements engineering for services. In: Proceedings of the 5th IEEE International Conference on Research Challenges in Information Science [RCISI. 19-21