A Methodology For Modeling Context Aware Services Based On Non-functional Requirements

Rachik Zineb1, Abdelbaki Issam2, Rahid Oulad Haj Thami3, Labriji El Houssine4
1Faculty of Sciences Ben M'SIK, Department of mathematics and informatics, Casablanca, Morocco
2Faculty of Sciences Ben M'SIK, Department of mathematics and informatics, Casablanca, Morocco
3National School of Computer Science and System Analysis (ENSIAS), Rabat, Morocco
4Faculty of Sciences Ben M'SIK, Department of mathematics and informatics, Casablanca, Morocco

Abstract

Service-Oriented Architectures (SOA) paradigm is mainly based on creating new software systems by loosely coupling services over a network. One accurate implementation of SOA is Web services. A web service is a logically atomic component that provides operations throw a standardized interface and is accessible via the internet. Web services are also important because they are based upon open internet standards for description and invocation. In addition the use of mobile devices is in an incredible increase and it enables users to access services from any location whenever they want.
The convergence of mobile technologies and service paradigm has promoted the birth of a new design and development paradigm known as Context-Aware Service (CAS).a context aware system can offer pertinent services for their users depending on their context. On the other hand, Service-oriented systems (SOSs) promise high flexibility, improved maintainability, and simple re-use of functionality. Achieving these properties requires an understanding not only of the individual artifacts of the system but also their integration. In this context, non-functional aspects (or quality requirements) play an important role and should be integrated in the early phases of the development process of SOS or even CAS in order to develop reliable services. In this paper, we discuss modeling of non-functional aspects of context aware service-oriented systems.

Keywords: *Non-functional requirements, context aware services, modeling, context.*

I. INTRODUCTION

With the growing interest of developing high level models and paradigms, a range of domain-specific languages and standards are already available for engineering service-oriented architectures (SOAs), such as WSDL, BPEL, WCDL, WS-Policy and WS-Security. These deal with the various artifacts of SOA systems, such as service descriptions, orchestrations, policies, and non-functional properties at specification level.
However, expressing non-functional requirements within the development of service-oriented systems (SOSs) is still in its infancy. Current works have been challenged by ensuring reliable services because non-functional requirements are both related to the business rules of the application and the technical description of the infrastructure where the application is executed.
Dealing with non-functional requirements is not an easy task mainly because the methodology treating them must ensure enough abstraction for modeling quite different non-functional requirements and must lead to the implementation of protocols that ensure these properties while the application is executed. The other main problem in developing reliable services is the choice of non-functional requirements to be treated. Most of the times, we consider that this choice is not automatic and is left to the last phases of service development when the user preferences become visible, but again this method doesn't fill with integrating non-functional requirements in the early phases of services development.
In this paper, we use user context as a major clue for selecting the appropriate non-functional requirements we deal with while developing service oriented applications. In this optic our work is mainly based on modeling reliable context aware applications.

The remainder of this paper is structured as follows. In the next Section we highlight some preliminary notions. In section 3 we give an overview of related works. In Section 4, we present our non-functional requirements model and give the details of it, and we present, in Section 5, our selection approach.
Finally, in Section 6, we conclude with a summary of our contributions and the future perspectives of this work.

II. PRELIMINARY AND BACKGROUND

A. Service oriented architecture

SOA is an architectural style that promotes sharing and reusing software components (i.e., published services). Services are discoverable as service providers publish their services' descriptions in registries. Service consumers can then discover, select, and invoke or compose these published services to meet their business needs.

SOA is not tied to any specific technology and does not rely on any particular implementation, although it is commonly implemented using web services, being primarily developed by the Organization for the Advancement of Structured Information Standards (OASIS) and the World Wide Web Consortium (W3C).

As defined in the OASIS reference model for Service Oriented Architecture, "SOA is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations" [1]. The part emphasizing "under the control of different ownership domains" is of a particular interest. It implies the ability to use services provided by third parties, which often speeds up applications' development time in comparison with present long-established, tightly-coupled, embedded environments. However, when service providers and service consumers are not within the same organization, service descriptions could be the only means to "communicate". Service consumers do not have to develop or even understand the underlying logic and implementation details of services they use. Services abstract form their underlying logic, which means they share nothing but a formal contract that contains only the information required by service consumers to determine whether a given service is appropriate for their needs (including functional and some of non-functional properties of the service) and the information necessary to interact with the service such as service interfaces, behavior, and location.

User context

Context is the information that characterizes the interactions between humans, applications, and the environment [2]. several context definitions were proposed in the literature (e.g., [3, 4]) serving various domains, however the context definition given by Dey and Abowd remains the most generic. In fact, these authors have defined context as "any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves" [5]. As given in [6], we consider context parameters as any additional information that can be used to improve the behavior of a service in a situation. Without such information, the service should be operable as normal but with context information, it is arguable that the service can operate better or more appropriately [7].

In our work, we consider the context specification given in [8] as they have given a context metamodel that is abstract, generic and extendable. They consider the context as a set of parameters (e.g., language, location) and entities (e.g., user, device) that can be structured on sub contexts. Sub contexts can also be recursively decomposed into categories. Context may be constituted of simple parameters (e.g., language), derived parameters (i.e., computed from other parameters; for example: a distance parameter can be computed from two GPS positions) and complex parameters (e.g., location) which have representations (e.g., DMS and DD representation for the location parameter)

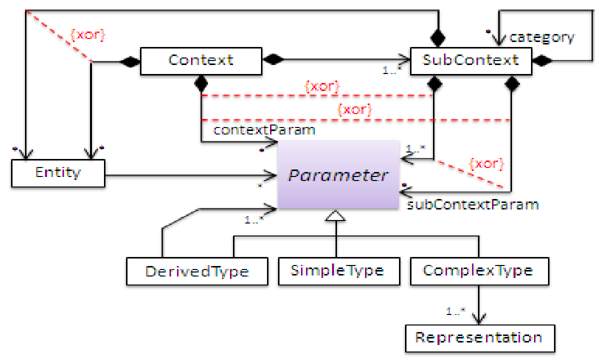

Figure 1. Core context metamodel.

III. RELATED WORKS

A. NFRs and Qualities in SOA

Nowadays, there is still no complete standardized solution specifying what service providers should expose as NFP information in their service descriptions. We stress the fact that the focus is not on all the NFPs of the SOA service but only the subset of the NFPs of interest to the service consumers: the consumer's perspective.

In the literature, non-functional requirements are often referred to as qualities of an application. Other terms often used for non-functional requirements include constraints, quality attributes, quality goals, and quality of service requirements.

In our work we have considered when extracting our initial set of NFPs includes the following:

- The Quality Model part of the ISO/IEC 9126 [9], which is an international standard for evaluating the quality of developed applications (i.e., non-SOA). It is thought defined for traditional software engineering, and is not entirely relevant for SOA, where the NFPs are often defined without involving the service consumers.

- The Architecture Working Group of the W3C Web Services working on the architecture of web services, has identified a set of QoS parameters for web services [10] , namely : performance (throughput (throughput Eng.), the response time and execution time , reliability, scalability and adaptation of scale, capacity, robustness, exception processing factor , accuracy , integrity , accessibility , availability, interoperability , security, and QoS requirements related to the network . Most researches that attempted to identify and classify QoS parameters have considered parameters defined by the W3C which are associated with other parameters.

- QoS model for web services proposed in [9] , suggests a primary classification of QoS attributes based on independent environmental attributes of the service (functional part) and attributes depend on the environment of the service(non functional part).

- The work in [11] identified and organized the QoS attributes of Web services into categories:
  - Attributes related to the execution: scalability , capacity, performance (response time , latency , and throughput) , reliability, availability , strength / flexibility, exception processing , and accuracy .
  - Attributes related to transaction support: Integrity, transaction.
  - Attributes related to price and configuration management: Standard supported stability, price and completeness.
  - Attributes related to security: authentication, authorization, confidentiality, traceability, data encryption, and non-repudiation.

- The Quality Model Working Draft (WS-Quality factors) [12][13], initiated in September 2005 by OASIS with the aim to define NFPs specifically for web services technology. The authors explain guidelines and detailed implementation notations for the service provider. It aims to define and formalize a set of attributes in the context of contracting for web services. It is tied to web service technologies and is not focused on service description from the perspective of the service end consumers.

- Tosic et al. [14] define a Web Service Offerings Language (WSOL) to describe different classes profiles of service in a formal way (e.g., services refer to the same WSDL but have different levels of QoS constraints). A service is defined as various formal constraints (e.g., functional constraints, QoS, and access rights). Tosic et al. confirm the importance of NFPs in service selection and composition, and they developed mechanisms to monitor and adapt NFPs to enable dynamic service management. However, WSOL does not define QoS constraints to be monitored but uses external ontologies of NFPs.

- Choi et al. [15] have identified some of the unique features of SOA and then derived six quality attributes and their respective metrics to measure each attribute. The proposed set of attributes is intended to be used by service providers to ensure that a qualified service is published. This is a focus different than ours.

- Glaster et al. [16] recognize the critical importance and the difficulties associated with handling NFPs in general and the fact that they are even more difficult to handle in the SOA context. They attempt to generate a checklist of NFPs for SOA to be used by the service providers. Their NFPs are organized into three categories: process requirements, non-functional external requirements, and non-functional service requirements. The authors provide an extensive generic list of various NFPs to be considered when services are under development. However, they provide only informal information on the measurability of these NFPs.

- Balfagih and Hassan [17] examine the quality of service (QoS) of SOA and web services, and classify them from the perspectives of the developer, provider, and consumer. Their list of QoS from the perspective of the consumer includes response time, availability, reliability, security, usability, composability, and robustness. They define the three first QoS and propose corresponding metrics. For security, they just mention a few sub-factors including confidentiality, integrity, authentication, and availability. They define usability but again they do not propose any metrics; they simply mention sub-factors including understandability and configurability.

- O'Sullivan et al [18] are in line with our vision on the need for a basic set of domain-independent non-functional properties that can be used to improve discovery, comparison and service substitution. Nowadays, SOA-based services are becoming very convenient (e.g., in terms of price, speed and availability) when compared to conventional services. Their second concern raised is called the "Semantic myopia" and is defined as not taking advantage of the semantic richness of NFPs. We agree that NFP descriptions in SOA should take advantage of all the existing work related to non-functional requirements to address the new SOA challenges. The large technical report by O'Sullivan *et al.* [18] contains 79 models that describe NFPs covering the following: availability, payment, price, discounts, obligations, rights, penalties, trust, security, and quality. The authors have done a comprehensive work on the importance of NFP detailed descriptions as a motor to improve discovery, comparison and service substitution.

- Based on the work in [18], Becha et Amyot [19] have provided a catalogue of 17 relevant NFR as well as a concise definition and additional notes to justify their relevance, and describe how to measure them. This new catalogue is important because it focuses on the customer's perspective, it was validated with real users and experts, and it was defined in a domain-independent and technology-independent way so that it can be formalized and used by multiple developers and standardization bodies.

B. Methodologies for building reliable services

Over the last few years, a number of approaches have been proposed for the development of web services. Some address service composition: workflow definition or semantic equivalence between services and some propose new languages for service description.

There are many methodologies that address the service based development. The methodologies analyzed in this work are mainly the ones that address reliable services development: SOMF [20] representing the model based development for web services mainly; S-cube [21] dealing with business processes and service based development and PI SOD-M [22] a methodology based on MDA to develop reliable services .

- S-Cube: The S-Cube Framework [21] proposes an integrated structure for developing service-based applications. S-Cube offers three areas: Business Process Management, Coin-position and Coordination of Services, and Infrastructure. These areas are the backbone of the framework that are directly linked to three other areas for supporting systems development: Engineering and Software Design; Monitoring; and Security and Software-Quality.The methodology aims to guide the development of applications and describes some essential activities, such as (i) description of business objectives. (ii) domain assumptions defining. which are preconditions to be met for a particular application domain. (iii) description of domains, and (ii') description of scenarios for each domain.

  The S-Cube methodology does not provide an exhaustive list of rules for describing services. The S-Cube framework provides activities in various service-oriented development areas, however, it is still required to apply its concepts in real case studies to give an idea of its application, given the fact that its structure is very complex and multidisciplinary.
- SOMF: Service—Oriented Modeling Framework [21] is a model oriented methodology for modeling software with the best practices of software project activities and different architectural setting. SOMF can be used to describe enterprise architectures, service-oriented architecture (SOA) and cloud computing. SOMF offers a variety of modeling practices and disciplines that can contribute to developing successful service-oriented applications. The main goals of SOMP are [20]:

  1. A methodology describes SOMF modeling activities and each model transformation:

  2. The diagrams are created obeying some project patterns.

  The methodology's model transformations in SOMF aim to describe and refine aspects in the system development process. The models are: discovery model, analysis model, design model, architectural model, implementation model, quality model, operations model, business model, governance model.
- PI SOD-M : it is an MDA (Model Driven Architecture) based methodology. It provides an environment for building service compositions considering their non-functional requirements.

  PI SOD-M extends the SOD-M [23] method by adding the concept of Policy [24, 25] for representing NFR associated to service-based applications. PI SOD-M also proposes the generation of a set of models at different abstraction levels, as well as transformations between these models.

  PI SOD-M's models represent both the cross-cutting aspects of the application being modeled, as well as the constraints associated to services. The systems cross-cutting concerns affect functional concerns, such as availability: recovery: and persistence aspects. Constraints are restrictions that must be respected during the execution of the application, for example the fact that a service requires an authentication for executing system functions.

IV. NON-FUNCTIONAL CONTEXT AWARE SERVICE MODELING APPROACH

A. Modeling non-functional properties

The approach we propose in this paper defines a model where non-functional properties become first-class entities beside functional properties. The non-functional requirements; also named quality factors; describe characteristics the service must satisfy. Unlike any other approach, we propose to consider user context while adapting services to user's quality preferences.

The process is divides into two parts or integrated fields: i) Non-functional property (NFP) Modeling; ii) non-functional services modeling.

Several attempts have been made to define common non-functional property definition formalism, but the most promising ones are those that propose a generic meta-model for the definition of NFPs. We believe that user context is useful while defining NFP as one NFP can be proprietary in a specific context but not in another. Given the example of a tourist with a low battery mobile looking for a suitable restaurant, the service dynamically adapt itself into making execution time and reliability more proprietary than security and authentication. The same tourist with a higher level battery mobile will have the same service of searching restaurants; only the service this time adapt itself into making security and efficiency number one priorities now that the battery allows it. In this example the battery level is context information that defines the priority level of non-functional properties.

In the sub-process of NFP modeling, the non-functional properties include quantitative and qualitative priorities. The former has to specify the metrics used to quantify them as for qualitative properties cannot be measured in general.

A non-functional priority also have a weight, it's a way to specify its importance degree in a given context. It also has attributes that are a group of semantically correlated NFP. For example, security is an NFP that comprise attributes such as confidentiality and integrity. These attributes are instantiated and attached to any model element of the system model in order to express new aspects of non-functional characteristics.

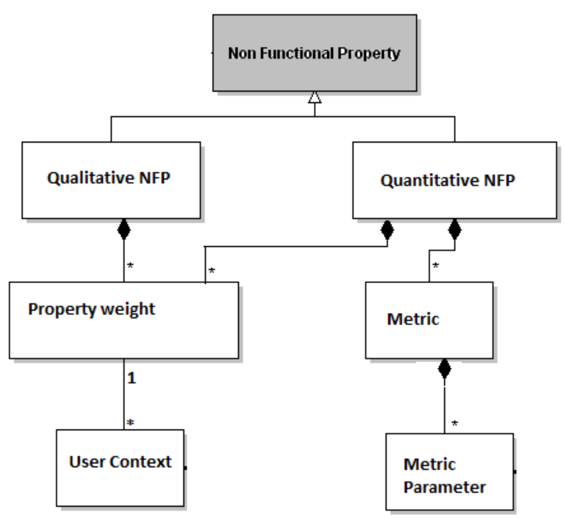

Figure 2. A non-functional meta-model.

B. Modeling reliable context aware services

The methodology relies on the key concept of service adaptation. An adaptation is the process of adjusting a service into a reliable service in a given context. The figure bellow is a meta-model proposed by [25]

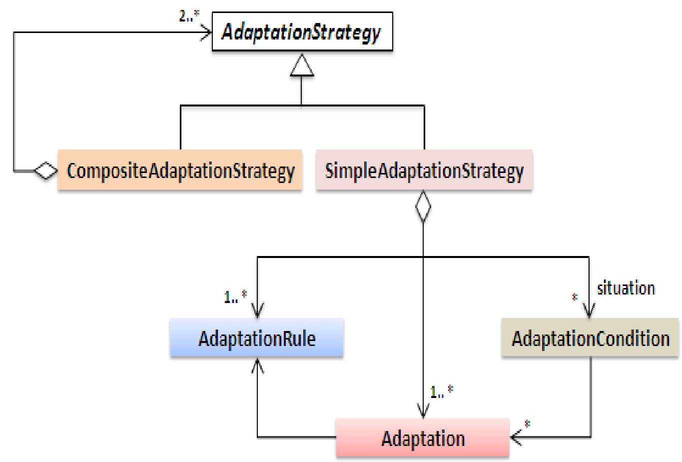

Figure 3. An adaptation strategy meta-model.

An adaptation strategy can be composite or simple; it aggregates a set of strategies that are constrained by rules and conditions. This meta-model was proposed in the aim of adapting the functionalities of service to its current context. We use it in our work in order to adapt it to the user's non-functional requirements as well.

In our modeling process, a service is modeled to be adaptable to the final user's quality requirements. The quality factors to be taken into consideration and their priority are up to the final user. This has inspired us to use the concept of user context.

The adaptation of a service to the user's NFR is achieved within an adaptation strategy. The elements in our model elements will be explained as follows:

- 'Service'
- 'UserContext': relevant information of the execution context. We believe that QoS factors considered while selecting services can change depending on their users contexts.
- 'NFFV': The non functional factors vector. It is the initial set of QoS factors.
- 'WCCV': The weighting coefficients context dependent vector. The weighting coefficients vector allow us to assign weights to the non functional factors in the 'NFFCV', based on the user context as well.
- UC-AdaptationStrategy: the user context adaptation strategy is an adaptation strategy that concerns a specific user context.
- AdaptationStrategy: an adaptation strategy is based on a rule that specifies the place in service that has to be modified.it aggregates a set of user context adaptation strategys.
- NFR-AdaptableService: a non-functional requirements adaptable service is a service with an adaptation strategy that concerns different user contexts.

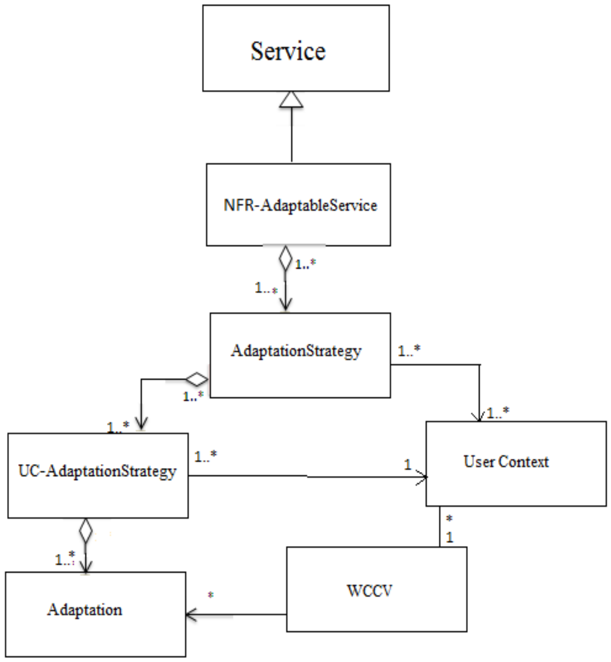

Figure 4. NFP-Adaptable services meta-model.

Therefore, a reliable service is a service that adapts dynamically to users' non-functional requirements.

For each user we associate a user context, and depending on the context we define the priority of non-functional requirements to be treated (weights).

For each service we associate an adaptation strategy (AdaptationStrategy) that is a set of adaptations(Adaptation) all based on non-functional requirements priority levels (WCCV). An ordered set of adaptations (Adaptation) will be applied on the basic service, to provide appropriate and reliable behavior in the current execution context. The matching result forms the NFR-AdaptableService.

I. Conclusion and Further Work

We have presented a method for developing reliable services, a reliable service being a service that satisfies users' nun-functional requirements.

Our method consists of a number of elements, notably a generic model for capturing non-functional attributes to be taken into account based on the user's context, and a method of dynamically adapting services into reliable services. The latter takes into account that some non-functional attributes are more preferential then others in some contexts.

Future work includes measurement of the non-functional attributes of services and using the user context to measure his non-functional requirements.

Another aspect for future work is enhancing the developing process to include services composition: services are usually not executed on their own but in the context of other services and hence one might make different choices depending on the usage environment (a cheaper product buying service might become less preferential if high shipping costs occur).

References

[1] OASIS, Reference Model for Service Oriented Architec-ture 1.0. October 2006. http://www.oasis-open.org/ specs/index.php#soa-rmv1.0 (last accessed: July 23, 2011)
[2] P. Brezillon, "Focusing on context in human-centered computing," IEEE Intelligent Syst., vol. 18, no. 3, pp. 62-66, May 2003.
[3] D. Salber, A. K. Dey and G. D. Abowd, "The Context Toolkit: aiding the development of context-enabled applications," Proc. SIGCHI Conf. on Human Factors in Computing Syst., Pittsburgh, PA, USA, May 1999, pp. 434-441.
[4] A. Schmidt, M. Beigl and H. W. Gellersen, "There is more to context than location," in Computers and Graphics J., Elsevier, vol. 23, no. 6, pp. 893-902, Dec. 1999.
[5] A. K. Dey and G. D. Abowd, "Towards a better understanding of context and context-awareness," GVU Center, Georgia Inst. of Technology, Tech. Rep. GIT-GVU-99-22, June 1999.
[6] H. L. Truong and S. Dustdar, "A survey on context-aware web service systems," Int. J. of Web Information Systems, vol. 5, no. 1, pp. 5-31, 2009.
[7] H. L. Truong and S. Dustdar, "Context coupling techniques for context-aware web service systems: an overview," in Enabling Context-Aware Web Services: Methods, Architectures, and Technologies, 1th ed.: Chapman and Hall/CRC, 2010, pp. 337-364.
[8]H. Hafiddi and Nassar, How Can Service Oriented Systems Make Beneficial
Use of Model Driven Architecture and Aspect
Paradigm?.
[9] ISO, Software Engineering - Product Quality - Part 1: Quality Model. ISO/IEC 9126-1, June 2001.
[10] OASIS, Reference Model for Service Oriented Architec-ture 1.0. October 2006. http://www.oasis-open.org/ specs/index.php#soa-rmv1.0 (last accessed: July 23, 2011)
[11] S.W. Choi, J.S. Her, and S.D. Kim, "QoS Metrics for Evaluating Services from the Perspective of Service Pro-viders", in IEEE International Conference on e-Business Engineering (ICEBE 2007). IEEE CS, pp. 622–625, 2007. doi:10.1109/ICEBE.2007.107
[12] E. Kim and Y. Lee, Quality Model for Web Services v2.0, Committee Draft, OASIS, September 2005. http://www. oasis-open.org/committees/tc_home.php?wg_abbrev= wsqm (last accessed: July 23, 2011)
[13] E. Kim et al., Web Services Quality Factors Version 1.0. Committee Draft 02, OASIS, July 2010. http://docs.oasis-open.org/wsqm/wsqf/v1.0/WS-Quality-Factors-v1.0-cd02.html (last accessed: July 23, 2011)
[14] V. Tosic, B. Pagurek, K. Patel, B. Esfandiari, and W. Ma, "Management applications of the web service offerings language (WSOL)". Inf. Syst. 30, 7, November 2005, pp. 564–586. doi:10.1016/j.is.2004.11.005
[15] S.W. Choi, J.S. Her, and S.D. Kim, "QoS Metrics for Evaluating Services from the Perspective of Service Pro-viders", in IEEE International Conference on e-Business Engineering (ICEBE 2007). IEEE CS, pp. 622–625, 2007. doi:10.1109/ICEBE.2007.107
[16] M. Galster and E. Bucherer, "A Taxonomy for Identifying and Specifying Non-functional Requirements in Service-oriented Development", in Proceeding 2008 IEEE Con-gress on Services (SERVICES 2008: Part 1), IEEE CS, pp. 345–352, 2008. doi:10.1109/SERVICES-1.2008.51
[17] Z. Balfagih, M.F. Hassan, and M.F. "Quality Model for the web services from multi-stakeholders' perspective", in In-formation Management and Engineering (ICIME'09), pp. 287–291, Kuala Lumpur, Malaysia, 2009. doi:10.1109/ICIME.2009.11.
[18] J. O'Sullivan, D. Edmond, D., and A.H.t. Hofstede, "For-mal description of non-functional service properties". Technical FIT-TR-2005-01, Queensland University of Technology, Brisbane (2005). http://www.wsmo.org/ papers/OSullivanTR2005.pdf (last accessed: July 23, 2011)
[19] Non-Functional Properties in Service Oriented Architecture – A Consumer's Perspective .Hanane Becha and Daniel Amyot.
[20] M. Bell. Service—Oriented Modeling (SOA): Service Analysis, Design, and Architecture.John Wiley. 2008.
[21] M. P. Papazoglou. K. Pohl, M. Parkin, and A. Metzger, editors. Service Research Chall(nges and Solutions tbr the Future Internet — S—Cube — Towards Engineering, Managing and Adapting Service-Based Systems, volume 6500 of Lecture Notes in Computer Science. Springer, 2010.

[22] Souza Neto. Plácido Antônio.A methodology for building service-oriented applications in the presence of non-functional properties / Plácido Antônio de Souza Neto. — Natal. RN. 2012.
[23] M. V. de Castro. Aproximnacíon MDA para el Desarrollo Orientado a Servicios de Sistemas de Intòrmacíon Web: Del Modelo de Negocio al Modelo de Comnposicíon de Servicios Web. PhD thesis, Universidad Rey Juan Carlos - Escuela Técnica Superior de Ingeniería de Telecomunicación. 2007.
[24] J. A. Espinosa-Oviedo. G. Vargas-Solar. J. L. Zechinelli-Martini, and C. Collet. Non-functional properties and services coordination using contracts. In Proceedings 01 the 2009 Ínternationial Database Engineering & Applications Symposium, IDEAS '09, pages 307—310. New York, NY. USA. 2009. ACM.
[25] J. A. Espinosa-Oviedo. G. Vargas-Solar, J. L. Zechinelli-Martini. and C. Collet. Policy driven services coordination for building social networks based applications. In In Proc.01 the 8th hit. Conference on Services Computing (SCC' II), Work—in—Progress Track,Washington, DC. USA, July 2011. IEEE.
[25] Hafiddi et al., 2011-d. H. Hafiddi, M. Nassar et A. Kriouile, "Rapport technique : Architecture
de référence et catalogue des services du projet ICTESAD", Rapport technique, CSPT/RT-30-11-2011-FR, Novembre 2011.