



Non-Functional Requirement-Based Service Ranking and Selection

RACHIK ZINEB¹, ABDELBAKI ISSAM², RACHID OULAD HAJ THAMI³, LABRIJI EL HOUSSIN⁴

^{1,2,4} Faculty of Sciences Ben M'SIK, Department of mathematics and informatics, Casablanca, Morocco

³ National School of Computer Science and System Analysis (ENSIAS), Rabat, Morocco

E-mail: ¹zineb.rachik@gmail.com, ²i.abdelbaki@gmail.com, ³oulad@ensias.ma, ⁴labriji@yahoo.fr

ABSTRACT

Since more and more applications want to use services that most accurately meet their requirements, we believe that non-functional requirements service selection mechanisms will play an essential role in service-oriented architectures. The integration of these non-functional requirements such as quality, safety, reliability, availability, location, price ... in the process of ranking and selecting services is currently the subject of several research works. Recently published solutions propose using optimization programming techniques. However, the poor scalability of program solving methods restricts their applicability to small-size problems and renders them inappropriate for dynamic applications. In this paper, in addition to a mathematical approach we choose to propose a non functional requirements-based model for selecting services which is, at the same time, generic and abstract. We also consider the notion of user context in order to predict his non functional requirements instead of requesting them, case of figure of most research works.

Keywords: *Services, Ranking, selection, context, model, Non functional requirements.*

1 INTRODUCTION

Enterprise practice witnesses a growing interest in the ad-hoc model for service architecture in the areas of accounting, supply chain management, finances, e-Science as well as in multimedia applications. With the growing number of available services the choosing problem becomes a decision problem on the selection of component services from a set of alternative services that provide the same functionality. So it leads to the issue of selecting the best services among a list of “Candidate services”, with the same functionalities. These services are, of course, different from one another in non-functional properties such as response time, availability, throughput, security, reliability, and execution cost, and are therefore different in terms of non functional properties.

Given an abstract representation of a service request and given a list of functionally-equivalent service candidates for each task in the composition request, the goal of service selection approach is to

find one service from each list such that the user's non functional requirements are satisfied.

Dynamic user environments cause some difficulties in service selection. As some of the non functional requirements should be taken into account in some user contexts and not in other, such as image quality that is more preferential if the user device has a high pictures definition, and not proprietary if not. Therefore the user context should be taken into account while selecting services based on their non functional factors.

In this paper, we present a service selection approach that copes with above issues. This approach consists of two phases. In the first phase, we provide a model to describe in an abstract level the link between user context and user non functional requirement. In the second phase we apply a mathematical method of classification to rank and select services.

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.

2 PRELIMINARY AND BACKGROUND

2.1 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.

2.2 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)

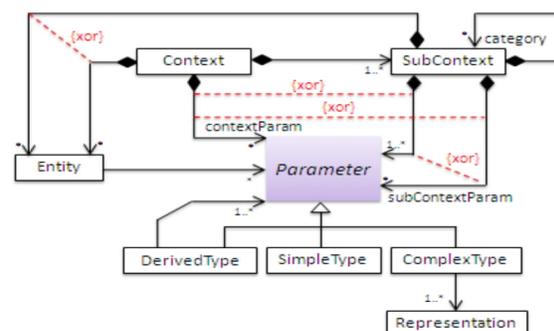


Fig. 1. Core context metamodel.

3 RELATED WORKS

3.1 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.

3.2 NFRs-based service ranking and selection

The QoS-based service ranking and selection in service-oriented applications has gained the attention of many researchers some approaches (e.g., [20,21]) use the benefits of Data mining algorithms in knowledge extraction and pattern discovery to deal with the huge amounts of data in web service selection.

Wu et al. [20] present a Bayesian network QoS based assessment model for web services. That could predict the service capability in various

combinations of users' QoS requirements. This approach is used to evaluate the capability of each service, and the one with best capability is selected as the binding service. Though it uses Bayesian network classification algorithm for each provider/service to predict the level of QoS, it is complex in computation terms and is based on probabilities, moreover it just considers local constraints in web service selection and doesn't mention the user's context.

Ben Mabrouk et al. [21] present a heuristic approach for service composition in dynamic environments. This solution uses the K-means algorithm to classify the web services to QoS levels, then it uses the result of this clustering in an utility function in order to rank web service candidates for each task as a local selection part, then it uses a search tree to select the best services to form the composition plans in an ordered way. The proposed solution is computationally expensive in both of the clustering algorithm and the structure of the search tree in a composition plans with the high number of activities. Furthermore it does not mention the relationship between NFP and the user context.

4 THE QOS CONTEXT LINKING MODEL

When thinking about adding further functionality to services selection through extra-functional properties, we have to consider how to do it whilst involving the user context. Unfortunately, there is no standard proposal for selecting services under user context constraints in the literature.

In order to cope with above issues, we propose to model the system whose elements will be explained as follows:

- 'Service'
- 'User': The end consumer (user) of the 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.
- 'NFFCV': The non functional factors context dependent vector. It is a set of the QoS factors to be considered in the ranking and the selecting process, based on the user context.
- '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.

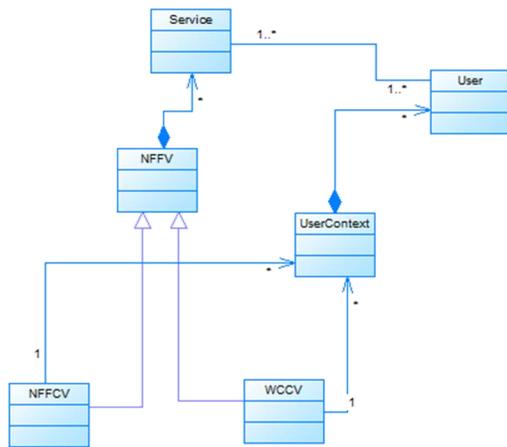


Fig.. 2. QF-context model.

We consider that each service has one ‘NFFV’ (non functional factors vector) based on our analysis of most relevant non functional factors. This initial vector is modified based on the user context.

In our approach we rank and select services based on their NFP. Having the user’s most preferential NFP thanks to its context enables our selection approach to be more efficient.

In the next section, we describe our mathematical approach of ranking and selecting services based on their NFP.

5 THE SELECTING APPROACH

Once the ‘NFFCV’ and ‘WCCV’ set, we can begin the second phase of our approach. In this work, we consider that metrics for non-functional properties are computed, and we focus on using these metrics to select services that accurate the user’s non functional needs the most.

Let $S_1..S_m$ be the services that meet the user’s request in functional terms. Our mission is to select from this set of services only those whom non functional properties are answering user’s non-functional requirements as well.

Let $Q [q_1...q_n]$ be the quality vector of a given service.

For each user, we consider the quality vector $V [v_1...v_n]$ corresponding to its non-functional needs, and $P[P_1..P_n]$ corresponding to the ‘WCCV’ that we have described in the previous section. The goal is then to select the services whom vectors $Q [q_1...q_n]$ are identical or close to the vector $[v_1, ...,v_n]$. And we consider the function ‘DIS’ that

computes the distance between two vectors $V [v_1...v_n]$ and $Q [q_1...q_n]$. Applied to a service $S [q_1...q_n]$ ‘DIS’ calculates its degree of approximation of the vector $[v_1... v_n]$.

The formula of the mathematical function ‘DIS’ is given by:

$$DIS (S) = (v_1- q_1)^2 + (v_2- q_2)^2 + (v_3- q_3)^2 +..... + (v_n- q_n)^2$$

When applying ‘DIS’ on each of the candidate services $S_1...S_m$, we get a set of values corresponding to distances between these services non-functional properties and user’s non-functional requirements. The service to select is then the one with the minimal distance. This first version of the ‘DIS’ function can be extended in a system where some user constraints are more preferential then others. In this case we use the weighting coefficients context dependent vector $P[P_1..P_n]$ and replace the ‘DIS’ function by formula: $DISP(S) = (q_1 - v_1)^2 * p_1 + (q_2 - v_2)^2 * p_2 + + (q_n - v_n)^2 * p_n$.

This way, we have built a global non-functional calculation function that reflects specific requirements and logic conditions. The logic conditions can be reflected by the value of the weight

6 CONCLUSION AND FURTHER WORK

We have presented a method for selecting services based on non-functional requirements adding to ongoing work on selection of services in general which mostly concentrates on functional aspects.

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 automatically ranking 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 ranking mechanism to also include ranking services that are a part of a service 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).

7 REFERENCES

- [1] OASIS, Reference Model for Service Oriented Architecture 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 Architecture 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 Providers", 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/wsrm/wsrf/v1.0/WS-Quality-Factors-v1.0-cd02.html> (last accessed: July 23, 2011)
- [14] V. Tasic, 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 Providers", 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 Congress 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 *Information 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, "Formal 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] G. Wu, J. Wei, X. Qiao, L. Li, "A Bayesian network based Qos assessment model for web services," *Proc. IEEE International Conference on Services Computing (SCC 2007)*, 2007, pp. 498-505.
- [21] N. B. Mabrouk, S. Beauche, E. Kuznetsova, N. Georgantas and V. Issarny, "QoS-aware Service Composition in Dynamic Service Oriented Environments," *Proc. Middleware*, 2009, pp.123-142.