# QoS-Aware Semantic Service Selection: An Optimization Problem

José María García, David Ruiz, Antonio Ruiz-Cortés, and Jose Antonio Parejo

Universidad de Sevilla
Dept. Lenguajes y Sistemas Informáticos
Escuela Técnica Superior de Ingeniería Informática
Av. Reina Mercedes s/n, 41012 Sevilla, Spain
Contact Email: josemgarcia@us.es

## Abstract

*In order to select the best suited service among a set of discovered services, with respect to* QoS *parameters, a user have to state his or her preferences, so services can be ranked according to these* QoS *parameters. Current Semantic Web Services ontologies do not support the definition of* QoS*-aware user preferences, though there are some proposals that extend those ontologies to allow selection based on those preferences. However, their selection algorithms are very coupled with user preferences descriptions, which are defined without semantics or at a different semantic level than service functionality. In this work, we present a service selection framework that transforms user preferences into an optimization problem where the best service is selected. This framework is based on an ontology that conceptualizes these user preferences. Thus, we use a very expressive solution decoupled with the concrete selection technique by using* XSL *transformations, while describing* QoS*-aware user preferences at the same semantic level of functional preferences.*

## 1. Introduction

Semantic Web Services (SWS) are becoming an important research area, with middlewares like WSMO [13] or OWL-S [9] being the most prominent ones. These middlewares or frameworks define an upper ontology to describe Web Services so they can be automatically discovered, composed, and invoked by means of user preferences. However, these approaches do not take quality of services (QoS) parameters into account when performing previous tasks, because they were developed focusing on describing functionality. Thus, previous tasks have been traditionally interpreted as a functional filter, where user preferences are matched with compatible provider preferences, in terms of their functionality descriptions, frequently using Description Logics (DLs) reasoners. Nowadays, the focus is on QoS-based tasks. However, these tasks lead to an optimization problem, where the best service among a set of services has to be selected, so DLs cannot be used in this context.

There are some extensions to current SWS frameworks that are aware of QoS, using different optimization techniques, such as [16], that introduce QoS parameter tendencies in WSMO that are manipulated within quality matrices; [18], that extends OWL-S by using fixed user preferences that are selected using matching degrees; or more generic approaches as [10] which does not handle QoS-aware user preferences natively, among others. However, their selection algorithms are very coupled with their proposed extensions, most of them being implemented ad-hoc. Thus, there is a semantic gap between functional user preferences (usually using WSMO or OWL-S) and QoS-aware preferences, which are specific for each proposal, using different ontologies or even non-semantic descriptions, and depending on its corresponding ad-hoc selection algorithm. It becomes necessary to provide semantics to QoS-aware preferences, such as an ontology that conceptualizes these user preferences, describing them in terms of utility functions, specified by the user [6].

Using the ontology extension proposed in [6], we present a generic selection framework that transforms user preferences definitions into an optimization problem, using different techniques to solve it, such as Constraint Programming, Linear Programming or Dynamic Programming. Due to this ontology instances are defined in XML, the most simple and powerful approach to perform these transformations is to use XSL style sheets [4,5,11]. Thus, an XSL transformation are used to obtain the optimization problem that represents a selection process using corresponding QoS-aware user preferences. Using this proposal, the selection technique is no longer coupled with the representation of QoS parameters, because of the use of a generic transformer. Furthermore, our solution enables the use of a common ontology

between providers and users, improving the automation of SWS tasks, such as discovery, selection, and composition.

This work is structured as follows. In Sec. 2 we briefly review some proposals on user preferences, QoS-aware selection of SWS, and XSL transformations applied to ontologies. Then, we introduce our ontology of QoS-aware user preferences and outline our proposed framework to perform the selection in Sec. 3. Section 4 sketches an actual mapping between our ontology and a Constraint Programming optimization technique. Finally, we show our conclusions in Sec. 5.

## 2. Related work

Concerning user preferences and utility functions, there are some proposals that uses them in selection tasks [8, 14, 17]. However, only Ruiz-Cortés *et al.* [14] allow the user to define complex utility functions. These proposals use optimization techniques, such as Integer Programming or Constraint Programming, to perform selection tasks. Therefore, utility functions emerge as an alternative approach to define highly expressive QoS-aware user preferences than weights [12] or QoS parameter tendencies [16], for instance.

Although there are many proposals that provide a semantic framework to define QoS [2, 3, 8, 10, 12, 16, 18], none of them define QoS-aware user preferences semantically. However, their ad-hoc, non-semantic definitions have an expressiveness varying from fixed preferences (cf. [2, 18]) to weights and QoS parameter tendencies (cf. [3, 12, 16]). Therefore, most of them perform selection tasks using descriptions at a different semantic level of functionality descriptions. In order to avoid these problems, QoS and user preferences have to be defined semantically [6].

On the other hand, QoS-aware user preferences can be transformed into an optimization problem, so selection are performed by a proper optimization engine [7]. Euzenat shows in [4] an API for ontology alignment that makes use of XSL transformations. Furthermore, Omelayenko and Fensel show that XSL transformations can be used to transform and map ontologies, applying them in an integration scenario [11]. In fact, Fensel and Bussler also applied XSL transformations within their Web Service Modeling Framework [5]. Thus, XSL can be used to perform transformations between ontologies, expressed in OWL or RDF, and to translate these ontological descriptions into an optimization problem.

## 3. Selection as an Optimization Problem

In a SWS selection scenario, we have a set of services descriptions that provide some functionality preferred by a user. From this set, selection process has to determine the best service in terms of QoS-aware user preferences. Thus, selection has to be treated as an optimization problem, transforming the input of this process (service descriptions and user preferences) into some representation that a concrete optimization technique can work with.

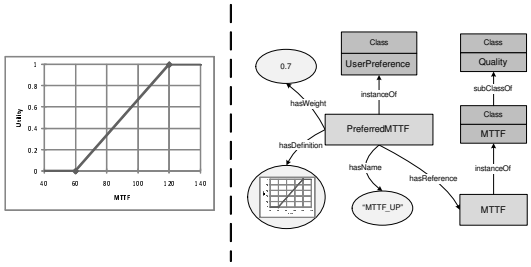


**Figure 1. An utility function and its user preference conceptualization.**

Our proposal provides a way to decouple the concrete selection technique chosen from the user preferences representation, providing a conceptualization of them. This conceptualization provides an ontology that adopts utility functions to define QoS-aware preferences. An utility function is defined as a normalized function (ranging over $[0, 1]$) whose domain is a QoS parameter, that gives information about which values of that QoS parameter are preferred by the user. Fig. 1 shows an example of a utility function for the mean time to failure ($MTTF$) parameter, along with an instance ontology that conceptualizes that function.

Each utility function has an associated weight that express how important (or preferred) is the corresponding QoS parameter for the user. Thus, an user preference is modeled as a composition of utility functions along its associated weights. That QoS-aware user preference is described as an instance of our user preferences ontology [6]. In that way, a user can compose utility functions into a global user preference, so he or she can express his or her preferences in terms of several QoS parameters, provided that the sum of the associated weights is equal to one.

The general form of this global user preference $\mathcal{UP}$ is as follows [14]:

$$\mathcal{UP}(p_1, \ldots, p_n) = \sum_{i=1}^{n} k_i \, U_i(p_i) \qquad k_i \in [0, 1] \sum_{i=1}^{n} k_i = 1$$

where each $p_i$ denotes a QoS parameter, each $k_i$ its associated weight ranging over $[0, 1]$, and each $U_i$ its associated utility function also ranging over $[0, 1]$. Thus, taking the semantic definition of each utility function along its associated weight, we can obtain the global user preference that has to be transformed into an optimization problem. With this conceptualization, the instances of the user preferences ontology can be transformed into a specification used by

a concrete technique, like Constraint Programming or Dynamic Programming.

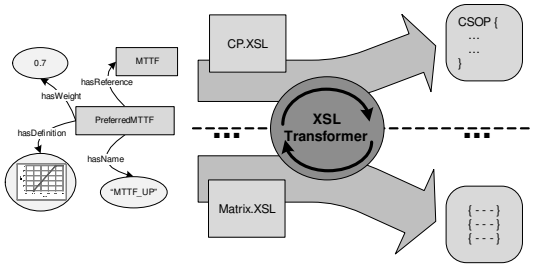


**Figure 2. Transforming user preferences into optimization problems.**

Our proposed framework performs the following steps. Firstly, the QoS parameters are extracted from the service semantic descriptions of the set of services which are going to be processed by the selection engine. Secondly, these descriptions are linked with user preferences that define utility functions for the corresponding QoS parameters. Finally, we apply an XSL transformation to these utility functions in order to obtain the specification of an optimization problem, which can be used as an input to a concrete solver. These transformations are summarized in Fig. 2.

Depending on the technique that is going to be used to perform the actual selection, a corresponding XSL style sheet has to be chosen. Thus, on the first hand there can be an XSL style sheet that transforms QoS-aware user preferences into a Constraint Satisfaction Optimization Problem (CSOP), using the selection algorithm proposed in [14]. On the other hand, there can be another XSL style sheet that transforms the same preferences into a quality matrix that can be used as the input for the selection algorithm described in [16]. Furthermore, other selection techniques can be supported, just by developing additional style sheets.

## 4. From Ontologies to CSOPs

Focusing on the transformation into CSOPs, there is a need to explicitly define the mapping between our proposed ontology [6] and a CSOP definition. A CSOP is an optimization problem that consists on a finite set of variables with their corresponding domains (including the variable to be optimized), and a set of constraints that have to be satisfied by an assignment of values to those variables from their domains, in order to consider that assignment as a candidate solution. Thus, the solution of a CSOP is the assignment that optimize the value of the variable to be optimized.

An example CSOP definition is presented in this section using an OPL model [15] in Fig. 3. OPL is an optimization language that can be used to properly define a CSOP. Thus, let two service providers, *P1* and *P2*, offer a service with the same functionality. In order to select one of them, an user states preferences about the $MTTF$ parameter shown in Fig. 1 and the mean time to repair ($MTTR$) parameter, defining corresponding utility functions and weights to each of them.

Each of the service providers have to guarantee actual values or range of values for each QoS parameter to be used into the selection process. Thus, instances of the QoS ontology from Maximilien and Singh [10] can be used to define those range of values (*MTTF_P1* and *MTTR_P1* concepts of the instance shown in Fig. 3). In our example from this section, the range of guaranteed values by providers *P1* and *P2* are shown in Table 1.

| Provider | MTTF | MTTR |
|---|---|---|
| *P1* | $75 \leq MTTF \leq 120$ | $5 \leq MTTR \leq 8$ |
| *P2* | $90 \leq MTTF \leq 105$ | $8 \leq MTTR \leq 10$ |

**Table 1. QoS range of values for each service provider.**

Once both providers guarantees and user preferences are described within our proposed semantic framework, these descriptions have to be transformed into an optimization problem. In Fig. 3, the corresponding OPL model that computes the global user preference (the $UTILITY$ value of the model) with regards to provider *P1* is shown. In this figure each concept and data value of the instance ontology is associated with its corresponding OPL model excerpt, so a comprehensive transformation between them can be developed.

The mapping sketched in Fig. 3 takes the QoS parameters referenced by each user preference ($MTTF$ and $MTTR$ concepts) and define their ranges in their corresponding CSOP variables. The *name* value of each preference serves as the name of the variable that holds the computed utility value. Concerning utility functions, each preference has its own function definition using the OpenMath [1] XML mathematical language. These definitions are mapped separately in the CSOP model as constraints. Finally, the corresponding weights are normalized between 0 and 100, due to some limitations of OPL models, so the global user preference ($UTILITY$) can be properly defined.

The CSOP discussed computes the utility value for the *P1* service, so it is also necessary to compute this utility value for the *P2* service. That computation can be performed by changing the constraints that model the provider guarantees. Thus, the service whose utility value is the maximum is finally selected. Note that utility values of each provided service are computed using a *worst-case* scenario, because the actual values of QoS parameters cannot

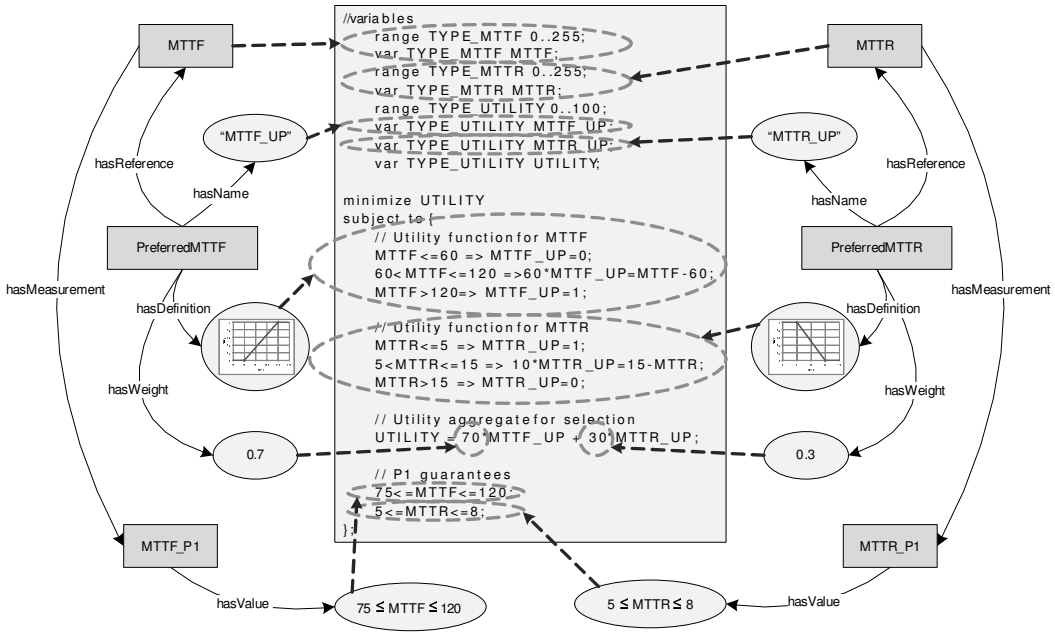


**Figure 3. Mapping between user preferences ontology instances and a CSOP.**

be guaranteed, so the $UTILITY$ computed value corresponds to the minimum of the global user preference [14].

## 5. Conclusions

User preferences have an important role in SWS selection. Using XSL transformations, QoS-aware user preferences can be expressed as optimization problems, which can be solved by different techniques, depending on the concrete XSL transformation used. An ontology to define these preferences is proposed, which can be used within any SWS framework, and is not coupled with the concrete selection technique used. Furthermore, a concrete mapping from our user preferences ontology to a CSOP is sketched in this paper as a motivating example.

Thus, QoS-aware user preferences are semantically defined at the same semantic level of functional preferences, by means of a generic ontology that can be linked with any SWS framework. Furthermore, these user preferences can be transformed into any optimization problem description, provided that we supply the corresponding XSL transformation that support the mapping from our ontology to the formalism of each optimization technique, enabling the extension of our framework.

## Acknowledgment

This work has been partially supported by the European Commission (FEDER) and Spanish Government under CICYT project Web-Factories (TIN2006-00472), and Andalusian Government project ISABEL (TIC-2533).

## References

[1] The OpenMath standard. Technical Report Version 2.0, The OpenMath Society, 2004.

[2] A. S. Bilgin and M. P. Singh. A DAML-based repository for QoS-aware semantic Web service selection. In *Web Services, 2004. Proceedings. IEEE International Conference on*, pages 368–375, 2004.

[3] G. Dobson, R. Lock, and I. Sommerville. QoSOnt: a QoS ontology for service-centric systems. In *EUROMICRO-SEAA*, pages 80–87. IEEE Computer Society, 2005.

[4] J. Euzenat. An API for ontology alignment. In *The Semantic Web - ISWC 2004*, volume 3298 of *LNCS*, pages 698–712. Springer, 2004.

[5] D. Fensel and C. Bussler. The web service modeling framework WSMF. *Electronic Commerce Research and Applications*, 1(2):113–137, 2002.

[6] J. M. García, D. Ruiz, and A. Ruiz-Cortés. On user preferences and utility functions in selection: A semantic approach. In *1st Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop*, LNCS. Springer, 2008. To appear.

[7] J. M. García, D. Ruiz, A. Ruiz-Cortés, O. Martín-Díaz, and M. Resinas. An hybrid, QoS-aware discovery of semantic web services using constraint programming. In B. Krämer, K. J. Lin, and P. Narasimhan, editors, *ICSOC 2007*, volume 4749 of *LNCS*, pages 69–80. Springer, 2007.

[8] K. Kritikos and D. Plexousakis. Semantic QoS metric matching. In *ECOWS 2006*, pages 265–274. IEEE Computer Society, 2006.

[9] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. Mcdermott, and Others. OWL-S: Semantic markup for web services. Technical Report 1.1, DAML, 2004.

[10] E. M. Maximilien and M. P. Singh. A framework and ontology for dynamic web services selection. *Internet Computing, IEEE*, 8(5):84–93, 2004.

[11] B. Omelayenko and D. Fensel. A two-layered integration approach for product information in B2B e-commerce. In *EC-Web*, volume 2115 of *LNCS*, pages 226–239, 2001.

[12] J. Pathak, N. Koul, D. Caragea, and V. G. Honavar. A framework for semantic web services discovery. In *WIDM '05: Proceedings of the 7th annual ACM international workshop on Web information and data management*, pages 45–50, New York, NY, USA, 2005. ACM Press.

[13] D. Roman, H. Lausen, and U. Keller. Web service modeling ontology (WSMO). Technical Report D2 v1.3 Final Draft, WSMO, 2006.

[14] A. Ruiz-Cortés, O. Martín-Díaz, A. Durán-Toro, and M. Toro. Improving the automatic procurement of web services using constraint programming. *Int. J. Cooperative Inf. Syst*, 14(4):439–468, 2005.

[15] P. Van Hentenryck. Constraint and integer programming in OPL. *INFORMS Journal on Computing*, 14(4):345–372, 2002.

[16] X. Wang, T. Vitvar, M. Kerrigan, and I. Toma. A QoS-aware selection model for semantic web services. In *International Conference on Service-Oriented Computing*, pages 390–401, 2006.

[17] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang. QoS-aware middleware for web services composition. *IEEE Transactions on Software Engineering*, 30(5):311–327, 2004.

[18] C. Zhou, L.-T. Chia, and B.-S. Lee. DAML-QoS ontology for web services. In *IEEE International Conference on Web Services*, pages 472–479, 2004.