On User Preferences and Utility Functions in Selection: A Semantic Approach

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

Universidad de Sevilla
Escuela Técnica Superior de Ingeniería Informática
Av. Reina Mercedes s/n, 41012 Sevilla, España
josemgarcia@us.es

Abstract. Discovery tasks in the context of Semantic Web Services are generally performed using Description Logics. However, this formalism is not suited when non-functional, numerical parameters are involved in the discovery process. Furthermore, in selection tasks, where an optimization algorithm is needed, DLs are not capable of computing the optimum. Although there are DLs extensions that can handle numerical parameters, they bring decidability problems. Other solutions, as hybrid approaches which use DLs in functional discovery and other formalisms in non-functional selection, do not provide a semantic framework to describe user preferences based on non-functional properties. In this work, we propose to semantically describe user preferences, so they can be used to perform selection within a hybrid solution. By using semantically described utility functions in order to define user preferences, our proposal enables interoperability between service offers and demands, while providing a high level of expressiveness in these preferences and including them within SWS descriptions.

Keywords: NFP-based Selection, Quality of Services, Utility Functions, Semantic Web Services.

1 Introduction

Concerning Semantic Web Services (SWS), discovery is one of the main research topics that have been widely studied and discussed, among others like composition. Description Logics (DLs) usually have become the natural choice when discovering SWS. Traditionally, discovery tasks have been interpreted as a functional filter, where demands are matched with compatible offers in terms of functionality. However, including non-functional properties (NFP) in the discovery process leads to an optimization problem. Selection of the best offer by means of their NFP has not been contemplated as a main task in discovering, so DLs reasoners are not well suited to select optimal offers. However, there are some proposals to perform NFP-based discovery, as the ones discussed in this work.

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

Optimization problems can be handled by solvers based on different formalisms, like Linear Programming, Constraint Programming, or Dynamic Programming, among others. Thus, it is possible to split discovery and selection tasks in terms of functional and non-functional requirements, so the former task can be performed by DLs reasoners, while the latter can be performed by solvers, taking a hybrid approach [3].

Focusing on selection, service demands have to state an optimality criterion, i.e. user preferences, so a solver can obtain the best offer in terms of these preferences. We propose to describe these user preferences by means of utility functions, whose domains are the different QoS parameters used to define NFP of service offers. In a SWS context, these utility functions have to be semantically described, allowing to match demands and offers described by different, but possibly equivalent, QoS parameters, enabling semantic interoperability between these descriptions.

The paper is structured as follows. In Sec. 2 we analyze current approaches on selecting SWS. Then, in Sec. 3 we show our proposal, describing what is a utility function and how to give semantics to it, showing an example. Finally, in Sec. 4 we discuss our conclusions.

2 Selecting Semantic Web Services

Once a set of services are discovered using a functional filter, the next step is to select the best offer in terms of NFP and user preferences. Thus, selection is modeled as an optimization problem. This kind of problem refers to a minimization or maximization of a real function, choosing the appropriate values of the involved variables.

In this context, the function to optimize is frequently called *utility function* or *objective function*. There are different techniques to obtain the optimal value of these functions, like Linear Programming, Constraint Programming or Dynamic Programming, for instance. In the following, we present the different approaches on selecting offers by means of NFP, characterizing their features and limitations with respect to user preferences.

2.1 Current Approaches

An early approach on modeling QoS in the context of SWS discovery are found in [10]. In this work, Ran presents a UDDI extension and a catalog of QoS parameters that can be included in UDDI descriptions. Discovery is performed using queries with functional requirements, as well as conditions on QoS. However, the actual discovery algorithm is not defined, and queries that use NFP are not shown, so their expressiveness is unknown. Additionally, UDDI only supports a keyword based search, so no form of inference or flexible match can be performed [15]. Apart from that, user preferences can not be expressed in the query and the resultant services are not ranked, so the user have to perform different queries in order to find the best suited service.

Although their proposal is not semantically defined, Liu *et al.* present a QoS computation model including a selection algorithm [5], which is adapted in other approaches [9,16]. They propose an extensible QoS model that comprises both generic and domain specific criteria. Selection is performed using an algorithm based on matrices normalization, where services are ranked in terms of their QoS matrix description and a vector of relative weights between QoS parameters, which express user preferences.

Pathak *et al.* also model mappings between ontologies in [9]. They propose to use domain specific ontologies to define NFP among offers and demands. In their work, selection is done using matching degrees at a first stage. Then, QoS parameters values are collected in a quality matrix, which is used to calculate a fixed, weighted utility function for each offer. Finally, offers whose utility function is above a given threshold, are ranked by one QoS parameter to obtain the optimal offer.

Wang *et al.* provide an extension to WSMO ontology [11] to handle QoS parameters [16]. They define a QoS selection model and an algorithm based on a quality matrix that contains values of QoS parameters. The user preferences are described in terms of tendencies, i.e. a demand may prefer parameters to be as small as possible, as large as possible, or around a given value.

Maximilien and Singh present a framework and a QoS ontology for dynamic selection in [8]. They use an agent-based approach where NFP are modeled via a three-layer ontology: an upper ontology which defines basic concepts associated with a quality parameter, a middle ontology which defines the most frequent QoS parameters and metrics, and a user-defined lower ontology that depends on the domain of the service. Although it constitutes a well-defined framework to semantically describe NFP and it is referenced by many authors [2,4,9], it lacks of a way to semantically describe user preferences.

An extension to DAML-S [1] to include QoS profiles is proposed in [18] by Zhou *et al.* This proposal only allows order conditions between QoS parameters, so it performs discovery and selection using DLs. The QoS ontology is simple and can be easily linked to the DAML-S service profile. However, its selection algorithm uses matching degrees to order the resulting set of services, so the user preferences can not be expressed, as they are inherent to that selection algorithm.

Another DAML-based proposal is also presented in [14], where S. Bilgin and Singh provide a DAML-based query language, instead of just extending OWL-S. Using this Semantic Web Services Query and Manipulation Language, they advertise QoS attributes and perform the selection. The main drawbacks of this approach are the same as in [10], with limitations on the expressiveness of queries, due to the use of DAML as its foundation. Thus, user preferences can not be expressed in those queries, and are inherent to their selection algorithm, as in [18].

Dobson *et al.* presents QoSOnt in [2], which is an ontology that extends OWL-S to describe QoS attributes and metrics. However, they do not explicitly explain how to perform selection, and their proposal suffers from OWL limitations, so

[1] DAML-S is an early version of OWL-S [6].

they have to use an ad-hoc XML language to allow custom data ranges. User preferences are modeled using the acceptability direction, that is the preferred tendency of metric values (e.g. the higher the best).

On the other hand, Zeng *et al.* show a basic QoS model to Web services composition in [17], although it can be applied to discovery and selection. They propose an algorithm based on utility functions, which are already defined for all the contemplated QoS parameters. The optimization is implemented using Integer Programming, providing weights to the different QoS parameters involved. The main drawbacks of this proposal are that it do not take semantics into account and that the utility functions are fixed, so the user can define its preferences only by means of weights.

Ruiz-Cortés *et al.* describe a QoS-aware discovery using Constraint Programming, where optimization is modeled as a Constraint Satisfaction Optimization Problem that minimize a weighted composition of utility functions, which are defined by the client using QoS parameters from a catalog [12]. As in [17], this proposal does not provide semantics, but user preferences, described by utility functions, can be defined by the user with high expressiveness.

An extension to [7] is presented in [4] by Kritikos and Plexousakis. They propose an ontology similar to the proposed by Maximilien and Singh [8], mixing offers and demands within an OWL-S description. Moreover, they present a matching algorithm to infer equivalences between different named QoS parameters that are semantically equivalent, although it is generally undecidable. Concerning discovery and selection, they use CSPs to perform the matchmaking of compatible offers, and then select the best service by means of a weighted composition of utility functions, which balance the worst and best scenarios to compute the utility value. However, these user preferences are not semantically defined in their QoS ontology.

2.2 Analysis

We show an analysis of the features of the different approaches introduced in the section before in Table 1. In this table, ordered by the order of exposition, we analyze if the given proposal semantically defines NFP, how it express user preferences, and the selection algorithm used.

We obtain several conclusions from this comparison. Firstly, there are a few proposals that uses utility functions to express user preferences [4,12,17], although only [12] allows the user to define complex utility functions. These three proposals use optimization techniques, as Integer Programming or Constraint Programming, to select the best offers. Therefore, utility functions become the natural choice to define highly expressive user preferences.

Secondly, there are many proposals that provide a semantic framework to define NFP [2,4,8,9,14,16,18], although [8] do not handle user preferences in their ontology and [14,18] have a fixed definition of user preferences, inherent to their selection algorithm. [4] is the most expressive when defining user preferences, followed by [2,9,16], that limit their preferences to weights and parameter

Table 1. Comparison between discussed proposals

Proposal	Semantic Defs.	User Preferences	Selection
Ran [10]	No	Not defined	Not defined
Liu et al. [5]	No	Weights	Quality matrix
Pathak et al. [9]	Yes	Weights	Quality matrix
Wang et al. [16]	Yes	Tendencies	Quality matrix
Maximilien & Singh [8]	Yes	External	Matching degree
Zhou et al. [18]	Yes	Fixed	Matching degree
S. Bilgin & Singh [14]	Yes	Fixed	Query lang.
Dobson et al. [2]	Yes	Tendencies	Not defined
Zeng et al. [17]	No	Utility and weights	Integer Prog.
Ruiz-Cortés et al. [12]	No	Utility and weights	Constraint Prog.
Kritikos & Plexousakis [4]	Yes	Utility and weights	Constraint Prog.

tendencies. According to all those proposals, it is clear that NFP have to be defined semantically.

Finally, we conclude that none of the above proposals semantically define user preferences, although in [2,16] the authors include in their ontology extension the tendency of QoS parameters. What is more, most of the proposals that perform selection tasks in terms of user preferences describe them using ad-hoc, non-semantic descriptions completely decoupled with the ones used to describe service functionality, causing a semantic gap between functional descriptions and user preferences.

The motivation of our work is precisely to tackle the previous problems. Most recent proposals use utility functions to express user preferences, and there are many NFP ontologies which our proposal can be integrated with. This proposal comes from mixing the expressiveness of utility functions and weights proposed by Ruiz-Cortés *et al.*, the semantic definition of NFP from Maximilien and Singh or Kritikos and Plexousakis, and an extension to give semantics to utility functions. Thus, we take full advantage of Semantic Web approaches on selecting the best services, while allowing to define user preferences using the most expressive solution, i.e. utility functions. Furthermore, we put functional, non-functional, and user preferences at the same semantic level, by means of using extensions to current SWS ontologies.

3 Our Proposal

Utility functions are the most expressive approach presented to describe user preferences that are used when selecting the best offers among a set. Although there are proposals that semantically describe QoS parameters and NFP, no one contemplates the conceptualization of utility functions. In this Section, we firstly give an overview of utility functions. Then, an ontology of user preferences is proposed to be used in the context of discovery and selection of SWS, showing a concrete example.

3.1 Utility Functions

An utility function is a normalized function (ranging over $[0, 1]$) whose domain is a given 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. This function is a piecewise linear function that defines a minimum utility (0) when $MTTF$ is below 60 minutes, and a maximum utility (1) when $MTTF$ is above 120 minutes. Between these two limits, the function grows linearly.

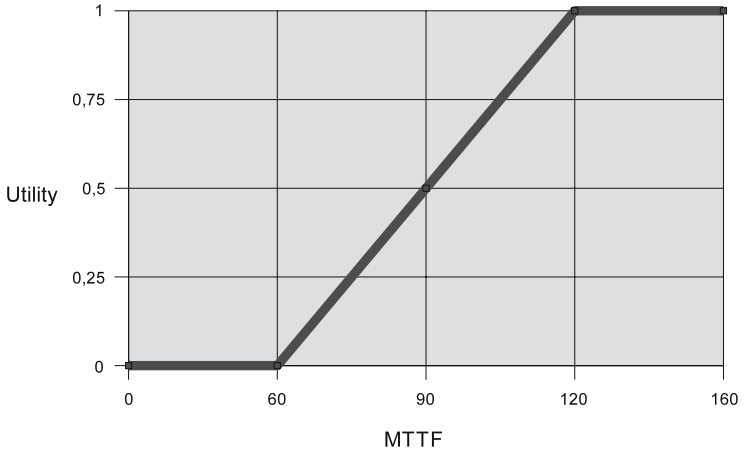

Fig. 1. Utility example for Mean Time To Failure

When selecting the best offers, a composition of different utility functions (one for each QoS parameter involved in NFPs) is often used to compute a global utility value, which serves to sort the service offers. In this composition, each utility function has an associated weight in the global function, so the user can specify how important is each QoS parameter when selecting offers. The general form of this weighted composition of utility functions $\mathcal{U}$ is as follows [12]:

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

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 $[0, 1]$ with the semantics previously defined.

3.2 Giving Semantics to Utility Functions

In order to provide semantic interoperability between utility functions defined on differently named QoS parameters, we propose to model these functions,

or more generally, user preferences, as an ontology. This ontology has to be instantiated by each user preference describing utility functions, so equivalences between QoS parameters can be inferred. Furthermore, this conceptualization allows the user to describe the whole service, including functional descriptions, at the same semantic level, without coupling user preferences descriptions with the selection algorithm.

Our proposed model is shown in Fig. 2. The main concept (or *class*) is UserPreference, which references a Quality concept via the *hasReference* object property. This Quality concept is analogous to the defined in [8], and represents the QoS parameter which is used in the definition of the corresponding user preference. Furthermore, the UserPreference concept has a key datatype property, *hasDefinition*, which links the more generic preference concept with the utility function that defines it. Note that Quality class is the link to QoS parameters used in the semantic definition of NFP. This definition can be performed using the ontology from Kritikos and Plexousakis [4] or from Maximilien and Singh [8], for instance.

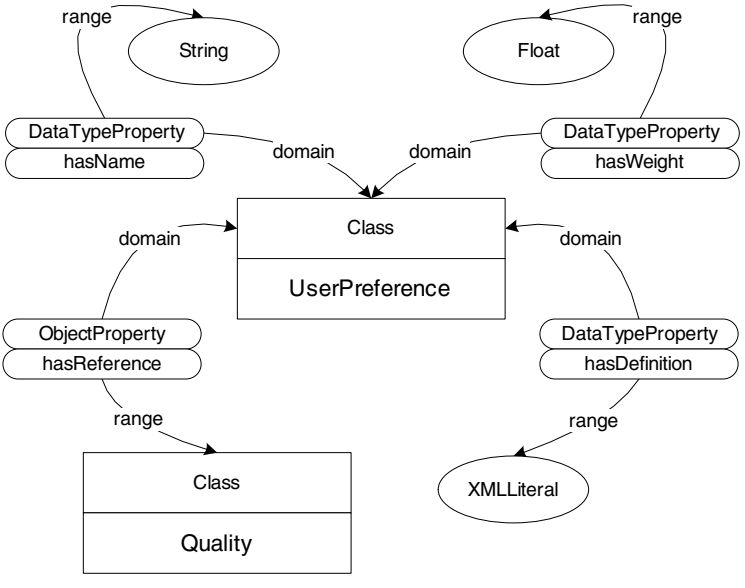

Fig. 2. Proposed ontology to model user preferences

The utility function is initially modeled as a property that contains an XML expression that describe the definition of each function in terms of OpenMath standard [1], as used in [4,13], allowing the evaluation of the function with a proper compiler or a mathematical tool, such as Mathematica.

Finally, our main concept UserPreference has two datatype properties: *hasName* and *hasWeight*. The former is used as an identifier of a given instance.

The latter is a real number which corresponds to the relative weight associated with the corresponding QoS parameter, used to compute the global utility function of an offer.

Fig. 3 shows an instance of our proposed ontology, in the case of an user preference about $MTTF$, with an associated weight of 1. Thus, the instance MTTFUserPreference references an instance UserMTTF of MTTF class, that is a subclass of Quality class from [8]. Moreover, the concrete utility function is specified as an OpenMath object that represent the one showed in Fig. 1, using XML.

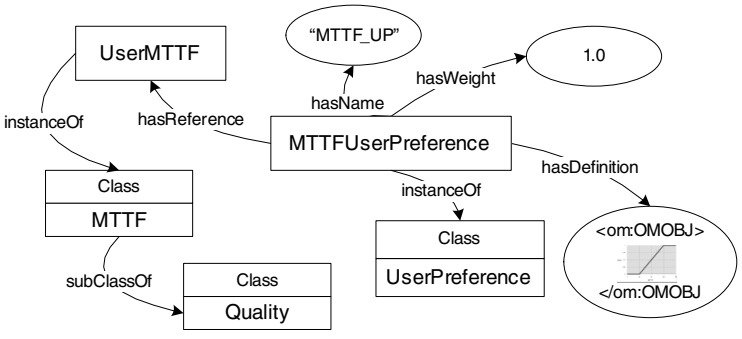

Fig. 3. Instance of our proposed ontology

The link with the rest of the semantic description of a service, including both functional and non-functional properties, is the QoS parameter $MTTF$, i.e UserMTTF instance in our example. In this case, the engine that perform the selection only has to be aware of the part showed in Fig. 3 and the corresponding NFPs that involve the $MTTF$ parameter, but generally, there are more parameters and user preferences involved in the selection process.

4 Conclusions

In this work, we provide a semantic framework to define user preferences on semantically defined QoS parameters, provided that it is used in conjunction with another proposal that semantically defines NFP, like [4,8]. Thus, all facets of SWS description (functional, non-functional, and user preferences) are described at the same semantic level, so discovery and selection tasks are completely done within a single semantic framework, allowing interoperability between different service definitions.

Furthermore, our proposal uses a very expressive solution to define user preferences, i.e. utility functions and weights, as in [4,12,17]. This formalism becomes more generic and powerful that ones used in other approaches. Additionally, we propose the use of a hybrid discovery engine to perform the different tasks in

discovery and selection using the best suited technique in each case. Thus, we optimize these tasks without compromising expressiveness.

In conclusion, our proposal allows a semantic definition of the whole discovery and selection process, using a hybrid approach without losing expressiveness. These facts allow to decouple the definition of user preferences from the concrete selection algorithm used.

Acknowledgments. The authors would like to thank the reviewers of the *NFPSLA-SOC'07 Workshop*, whose comments and suggestions improved the presentation substantially.

References

1. Buswell, S., Caprotti, O., Carlisle, D.P., Dewar, M.C., Gaëtano, M., Kohlhase, M.: The OpenMath standard. Technical Report Version 2.0, The OpenMath Society (2004)
2. Dobson, G., Lock, R., Sommerville, I.: Qosont: a qos ontology for service-centric systems. In: EUROMICRO-SEAA, pp. 80–87. IEEE Computer Society Press, Los Alamitos (2005)
3. García, J.M., Ruiz, D., Ruiz-Cortés, A., Martín-Díaz, O., Resinas, M.: An hybrid, QoS-aware discovery of semantic web services using constraint programming. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSOC 2007. LNCS, vol. 4749, pp. 69–80. Springer, Heidelberg (2007)
4. Kritikos, K., Plexousakis, D.: Semantic QoS metric matching. In: ECOWS 2006, pp. 265–274. IEEE Computer Society Press, Los Alamitos (2006)
5. Liu, Y., Ngu, A.H.H., Zeng, L.: Qos computation and policing in dynamic web service selection. In: WWW (Alternate Track Papers & Posters), pp. 66–73 (2004)
6. Martin, D., Burstein, M., Hobbs, J., Lassila, O., Mcdermott, D., et al.: OWL-S: Semantic Markup for Web Services. Technical Report 1.1, DAML (2004)
7. Martín-Díaz, O., Ruiz-Cortés, A., Benavides, D., Durán, A., Toro, M.: A quality-aware approach to web services procurement. In: 4th VLDB Workshop on Technologies for E-services TES 2003, pp. 42–53 (2003)
8. Maximilien, E.M., Singh, M.P.: A framework and ontology for dynamic web services selection. Internet Computing 8(5), 84–93 (2004)
9. Pathak, J., Koul, N., Caragea, D., Honavar, V.G.: A framework for semantic web services discovery. In: WIDM 2005: Proceedings of the 7th annual ACM international workshop on Web information and data management, New York, NY, USA, pp. 45–50. ACM Press, New York (2005)
10. Ran, S.: A model for web services discovery with QoS. SIGecom Exch. 4(1), 1–10 (2003)
11. Roman, D., Lausen, H., Keller, U.: Web Service Modeling Ontology (WSMO). Technical Report D2 v1.3 Final Draft, WSMO (2006)
12. Ruiz-Cortés, A., Martín-Díaz, O., Durán-Toro, A., Toro, M.: Improving the automatic procurement of web services using constraint programming. Int. J. Cooperative Inf. Syst. 14(4), 439–468 (2005)
13. Sánchez-Macián, A., López, D., López de Vergara, J.E., Pastor, E.: A framework for the automatic calculation of quality of experience in telematic services. In: 13th HP-OVUA Workshop, Sophia Antipolis, France (May 2006)

14. Soydan Bilgin, A., Singh, M.P.: A DAML-based repository for QoS-aware semantic Web service selection. In: IEEE International Conference on Web Services, pp. 368–375 (2004)
15. Sycara, K., Paolucci, M., Ankolekar, A., Srinivasan, N.: Automated discovery, interaction and composition of semantic web services. J. Web Sem. 1(1), 27–46 (2003)
16. Wang, X., Vitvar, T., Kerrigan, M., Toma, I.: A QoS-Aware Selection Model for Semantic Web Services. In: Dan, A., Lamersdorf, W. (eds.) ICSOC 2006. LNCS, vol. 4294, pp. 390–401. Springer, Heidelberg (2006)
17. Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H.: QoS-aware middleware for web services composition. IEEE Transactions on Software Engineering 30(5), 311–327 (2004)
18. Zhou, C., Chia, L., Lee, B.: DAML-QoS ontology for web services. In: IEEE International Conference on Web Services, pp. 472–479 (2004)