Louvain School
of Management

Working paper 08/04
Capturing and using QoS relationships to improve service selection

Caroline Herssens, Ivan J. Jureta, Stéphane Faulkner

Academic Year 2007- 2008

Louvain School of Management

Capturing and Using QoS Relationships to Improve Service Selection

Caroline Herssens[1], Ivan J. Jureta[2], Stéphane Faulkner[2]

[1] ISYS, LSM, Université catholique de Louvain, Belgium
[2] PReCISE, LSM, University of Namur, Belgium
caroline.herssens@uclouvain.be; iju@info.fundp.ac.be;
stephane.faulkner@fundp.ac.be

Abstract. In a Service-Oriented System (SOS), service requesters specify tasks that need to be executed and the quality levels to meet, whereas service providers advertise their services' capabilities and the quality levels they can reach. Service selectors then match to the relevant tasks, the candidate services that can perform these tasks to the most desirable quality levels. One of the key problems in QoS-aware service selection lies in managing tradeoffs among QoS expectations at runtime, that is, situations in which service requesters specify quality levels that cannot be simultaneously met. We propose a service selection approach that can deal with tradeoffs. The approach consists of: (i) rich QoS models to be used by service requesters when expressing QoS expectations and service providers when describing services' QoS, and for representing preference and priority relationships between QoS dimensions; and (ii) a multi-criteria decision making technique that uses the models for service selection.

Key words: QoS model, service selection, QoS relationships

1 Introduction

Engineering and managing the operation of increasingly complex information systems is a key challenge in computing. It is now widely acknowledged that degrees of automation needed in response cannot be achieved without open, distributed, interoperable, and modular systems capable of dynamic adaptation to changing operating conditions. Among the various approaches to building such systems, service-orientation stands out in terms of its reliance on the World Wide Web infrastructure, availability of standards for describing and enabling interaction between services, attention to interoperability, and uptake in industry. In a Service-Oriented System (SOS), service providers advertise the tasks that their services can perform and the Quality of Service (QoS) levels they can meet. Service requesters indicate tasks to execute and QoS levels to achieve. Service selectors (i.e., allocation mechanisms in SOS) then proceed to compare available services and select those that can execute the required tasks while achieving the most desirable feasible QoS levels.

Service selection is a fundamental issue in SOS because it determines how well the requests are satisfied [4, 12, 17, 28]. Comparing competing services (i.e., services that can execute the same tasks) over the levels of QoS dimensions they can meet is an appropriate approach to ensuring that quality expectations are met to the "best" feasible extent.

Contributions. Using QoS dimensions in service selection requires their definition by way of a QoS model [6]. The QoS model must cover all QoS constructs needed in service selection. Two models are needed in practice. Although both the service requester and service provider models must share the primitives for the representation of QoS dimensions and characteristics, and the definition of their values, the two models must also differ given the difference in purpose: the provider model should include relationships for stating dependencies between QoS dimensions while the requester model must involve relationships for defining requesters' priorities over QoS dimensions and preferences over the values of QoS dimensions. Given such models, the service selector will be able to appropriately compare services and take better selection decisions. Decision-making in presence of potentially many QoS dimensions and requests thereon, can be performed through multi-criteria decision analysis (MCDA). The aim is to rank competing services according to the values of their QoS characteristics. We propose an approach to service selection that responds to these considerations. The approach consists of:

- Rich QoS models used by service requesters when expressing QoS expectations and service providers when describing services' QoS, and for representing preference and priority relationships between QoS dimensions. The models are defined as extensions of the UML QoS framework metamodel [20].
- A multi-criteria decision making technique that uses the models for service selection. QoS models, and more precisely, the QoS relationships are used in a fuzzy MCDA approach to to build a fuzzy reference set on which interaction weights are set up, subsequently used for ranking competing services. To establish a ranking with fuzzy MCDA, the selector proceeds as follows:
  1. A reference set of service alternatives is built to compute weights of interacting criteria. The reference set is set up with help of requester's preferences and priorities and with providers' observed dependencies over QoS dimensions.
  2. Once weights are fixed, the selector applies them to the available services that correspond to the requested QoS dimensions and levels.
  3. Finally, values obtained with the application of fuzzy MCDA allow the selector to rank possible services and determine the optimal one to each service request.

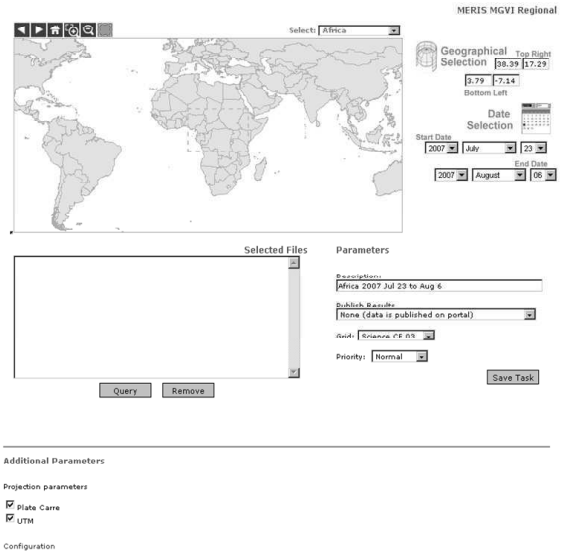
Organization. Section 2 introduces the case study used in the remaining of the paper. Section 3 assesses the UML QoS Profile with our proposed subdivision into distinct models and our relationships extensions while illustrating their utilizations through the case study. Relationships identified in QoS models are used

in Section 4 for an efficient selection based on fuzzy MCDA. This section also provides a selection example based on the case study. The Section 5 presents the related work of existing QoS models and exposes some existing selection approaches. Finally, Section 6 outlines conclusions and future work.

2 Case Study

In this section, we propose a case study subsequently used throughout the paper. The European Space Agency's (ESA) program on Earth observation allows researchers to access and use infrastructure operated and data collected by the agency.[3] Our case study focuses on the information provided by the MERIS instrument on the Envisat ESA satellite. MERIS is a programmable, medium-spectral resolution imaging spectrometer operating in the solar reflective spectral range. MERIS is used in observing ocean color and biology, vegetation and atmosphere and in particular clouds and precipitation. In relation to MERIS, web services are made available by the ESA for access to the data the instrument sends and access and use of the associated computing resources.

Fig. 1. Graphical user interface of the ENVISAT/MERIS MGVI web service



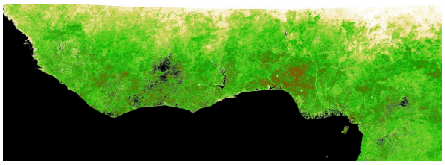
We are interested in the remainder about services able to provide vegetation indexes for a given region of the globe. A vegetation index measures the amount of vegetation on the Earth's surface. Below, we briefly review the functional requirements that the service satisfies. We then consider quality requirements.

Functional requirements. Services considered here process MERIS data and are able to extract the vegetation index. This processing can be selected for any time range (with the start of the satellite mission as the earliest time point); an option is available to delimit the region of the world of interest. The graphical

[3] http://gpod.eo.esa.int

user interface used to access the service is shown in Figure 1. Figure 2 illustrates the visualization of the output obtained for the Guinea - Cameroon region. The following are the required inputs of the service: *Time range*, *Bounding box* (to select a region of the globe), *Dataset*, *Publish site*, and *Projection type*.

Fig. 2. An illustration of the result provided by the ENVISAT/MERIS MGVI Web Service (Guinea - Cameroon region)



Quality requirements. Due to the calculations executed by the service and its parallel use, expected delays and availability are relevant quality considerations from the user's perspective. To make an appropriate selection, quality considerations need to be expressed by users and measured and advertised by providers. We focus on three such considerations, namely availability, reliability, and latency. For now we define them as follows. We then return to each throughout the paper and illustrate how our proposed extensions to the UML QoS profile work with this case study.

- Availability indicates the duration when a component is available for queries. Its value in percent is obtained as follows [23]:

$$A = \frac{upTime}{upTime + downTime}$$

- Reliability is a measure of confidence that the service is free from errors. Its value is given in percent and calculated as follows:

$$R = \frac{succeededAttempts}{succeededAttempts + failedAttempts}$$

- Latency measures the mean time taken by the platform to return the expected result. The value is given in minutes.

$$L = \frac{\sum_1^n networkTime + selectionTime + executionTime}{n}$$

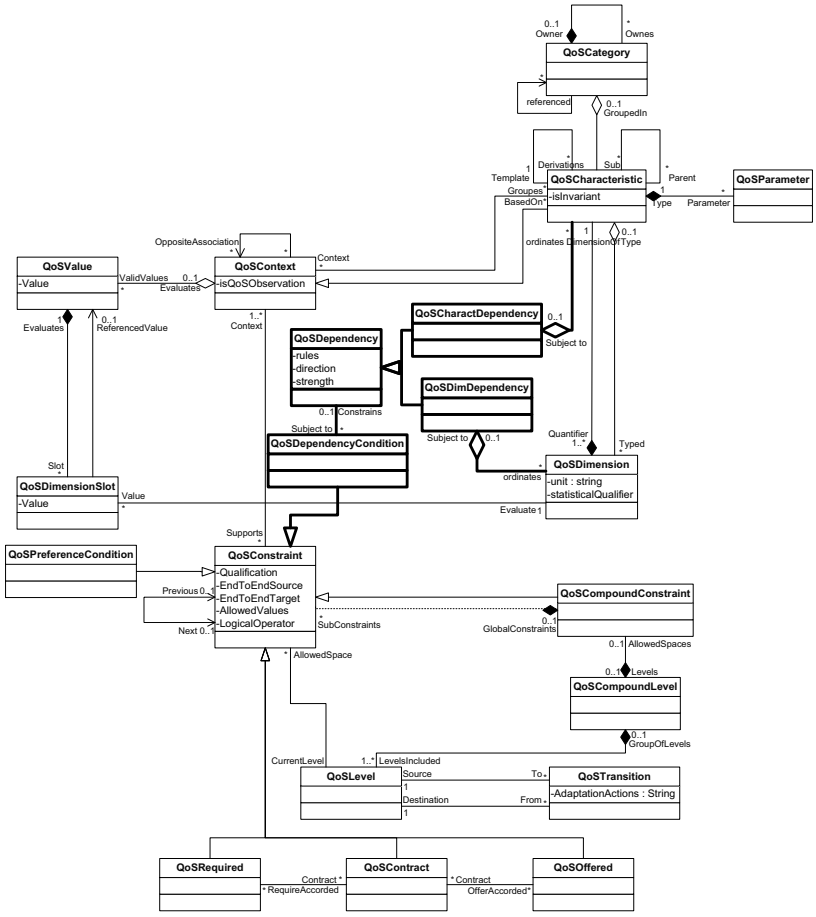
  where n is the total number of past executions. Latency of a such service is situated between 4 and 6 hours by day of the selected period due to the quantity of data to process. (This range is certified for a service requestor having network bandwidth of a least 15 mbits/s.)

This basic specification is incomplete. Further explanations are needed. A quality model will provide a checklist of relevant information and in this respect assist the requester and the provider in evaluating and managing the quality of the service.

3 Conceptual Foundations

This section overviews main concepts of the OMG UML QoS Framework and present our distinct models with our relationships extensions. Once instantiated, our QoS models are useful to lead service selection in Section 4.

Fig. 3. UML provider QoS Framework

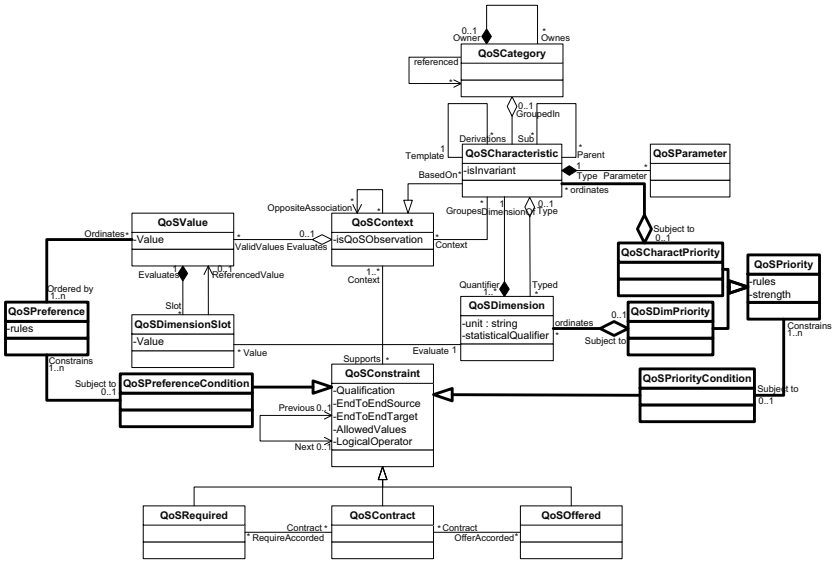


The UML QoS Framework metamodel introduced by the OMG in [20] includes different modeling constructs describing QoS concepts. It covers submodels aiming at defining different facets of QoS. The **QoS Characteristics submodel** outlines *QoS Characteristics* that are a description for some quality considerations and *QoS Dimensions* that are measures quantifying QoS Characteristics. *QoS Categories* are used to group together QoS Characteristics related to the same abstract quality topic. **QoS Constraint submodel** main constructs are *QoS Constraints* that restrict values of QoS Characteristics while stating limitations on modeling elements identified by application requirements and architectural decisions. The **QoS Level submodel** provides *QoS Levels*

that specify the working mode under which the service is executed. Complete description of constructs of these submodels is given in [20].

To make an explicit distinction between users requirements and providers advertisements, we split the OMG metamodel into two distinct metamodels. The service selector task will consist of match instantiation of the user metamodel with the one from the provider. The provider metamodel is illustrated in Figure 3 while the user metamodel is available in Figure 4. In addition to original modeling constructs, we have added some submodels aiming at express particular relationships existing over QoS Characteristics and QoS Dimensions:

Fig. 4. UML user QoS Framework

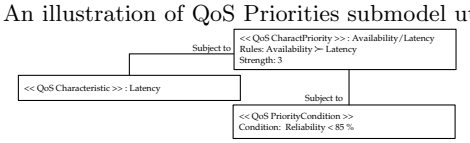


QoS Priorities submodel *QoS Priorities* are used to explicitly represent service requester priorities over QoS Characteristics and QoS Dimensions. Rules determine the order at which characteristics or dimensions are considered for optimization when services are being selected. The relative importance of the priority difference between elements is specified with the strength attribute . *QoS DimPriority* and *QoS CharactPriority* are specializations of QoSPriority defining specific elements for priorities over, respectively, dimensions and characteristics. *QoS PriorityCondition* are constraints specifying when priorities hold. The integration of these modeling constructs in the requester QoS model is illustrated in bold in Figure 4. An utilization of the QoS Priorities submodel is proposed in Example 1.

Example 1. In our case study, the requester of the service awards more importance to the *availability* characteristic than to the *latency* characteristic. This particular priority is constraint to a specific condition, stating that the priority is applied only if the *reliability* is inferior to 85%. This priority and its condition

are expressed with the UML user QoS Framework in Figure 5. This example will be used to build the reference set in Subsection 4.3.

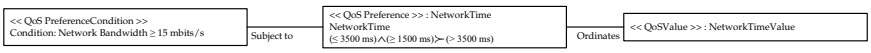
Fig. 5. An illustration of QoS Priorities submodel utilization



QoS Preferences submodel *QoS Preferences* enable the service requester to sort values of dimensions. Rules are used to determine a precedence order over values. The *QoS PreferenceCondition* indicates conditions for the preference on values to hold. This submodel is illustrated in bold on the user QoS model in Figure 4 and an example of utilization is given in Example 2.

Example 2. About the service providing regional vegetation indexes, the user has some preferences concerning the *networkTime*, he wishes that its value belongs to a specific range as illustrated in Figure 6. The instantiation of the preferences submodel also introduces a specific condition under which the preference is available.

Fig. 6. An illustration of QoS Preferences submodel utilization



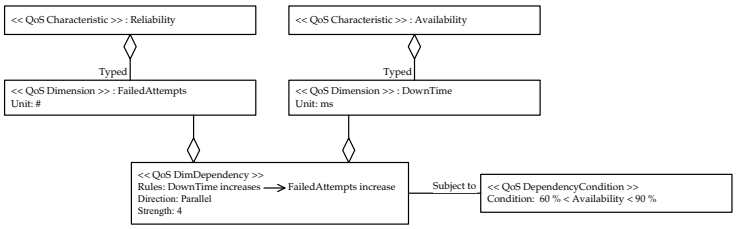
QoS Dependencies submodel *QoS Dependencies* allow to express explicitly dependency relationships existing over different QoS Characteristics or QoS Dimensions while specifying the strength and the direction of the link. The direction indicates that QoS Characteristics (respectively, QoS Dimensions) involved in the dependency are parallel or opposite, meaning that their direction is correlated or anti-correlated. The strength is represented with a level value between 1 and 10, corresponding to the importance of the correlation. The *QoS DependencyCondition* is used to define specific constraints under which a QoS Dependency is applicable. These extensions to the provider QoS model are available in bold in Figure 3 while their utilization is illustrated in Example 3.

Example 3. The provider of the MGVI/Regional service will use the dependencies submodel to underline interactions between several QoS Characteristics. In Figure 7, one of these dependencies is illustrated. *FailedAttempts* is one of the dimension used to quantify *reliability* while the *downTime* is a metric used to measure the value of the *availability*. The dependency appears on dimension level with the downTime inducing the number of failedAttempts. Moreover, as exposed in the instantiation of the submodel, this dependency is subject to a particular condition on the availability value.

4 QoS Driven Selection

Our aim is to provide a service selection approach that takes existing relationships identified over QoS Characteristics into consideration. In order to sort

Fig. 7. An illustration of QoS Dependencies submodel utilization



alternative services, we will use a particular class of methods of Multi-Criteria Decision Analysis (MCDA) [7]: the fuzzy MCDA. Fuzzy MCDA allows to establish a ranking over alternatives while accounting for multiple criteria, represented here by QoS Characteristics. Moreover, this technique introduces interaction indexes, able to express relationships over QoS Characteristics. The first step for the service selector is to build a reference set of alternatives that makes appear existing relationships over QoS Characteristics. Next, an algorithm is executed on the reference set in order to fix interaction indexes. Finally, the service selector calculates the score of existing alternatives with these interaction indexes and their ranking is established. Below, we outline how to obtain these indexes and how the ranking of available services is established with them.

In subsection 4.1, we introduce fuzzy MCDA concepts and explain how the utility value of each alternative can be calculated. In subsection 4.2, we describe how weights of criteria and interacting criteria are determined from a reference set established by the service selector and how provided services are then ranked. Finally, we present an example of utilization of our service selection approach in subsection 4.3.

4.1 Fuzzy Multiple Criteria Decision Analysis: concepts

Fuzzy MCDA methods [7] are of particular relevance in services selection because these allow to determine weights related to criteria but also weights related to interactions between criteria. Indeed, in case of interacting criteria, the usual weighted arithmetic mean ($U(x^a) = \sum_{i=1}^{n} w_i x_i^a$) is extended with a Choquet integral such that the utility of alternative a is calculated by $U(x^a) = \sum_{i=1}^{n} x_{(i)}^a \left[\mu(A_{(i)}) - \mu(A_{(i+1)})\right]$. With $a \in A$ which is the set of all possible alternatives and n as the total number of criteria considered. We can observe that the weights w_i which are considered as independent in the usual weighted arithmetic mean have been substituted by the weights $\mu(i_1, ..., i_k)$ in the extended mean. These weights, related to all possible combinations of criteria, make possible to express dependencies between criteria. For complete description of fuzzy measures, Shapley's values and Choquet integrals, see Marichal [15, 16] and cited references.

The overall importance of a criterion $i \in N$ is not solely determined by the weight $\mu(i)$ but also by all $\mu(S)$ such that $i \in S$, S being a subset of criteria related to the same subject. The importance index (Shapley value) of criterion i w.r.t. μ is defined by its Shapley's value, as in Equation 1.

$$\phi_{Sh}(i) = \sum_{T \subseteq N \setminus i} \frac{(n-t-1)!t!}{n!} \left[\mu(T \cup i) - \mu(T)\right] \tag{1}$$

To focus on interaction among subsets of criteria, the difference $a(ij) = \mu(ij) - \mu(i) - \mu(j)$ is used. The difference is 0 when the individual importances $\mu(i)$ and $\mu(j)$ add up without interfering. In this case, there is no interaction between criterion i and criterion j. If the criteria interfere in a positive way, the difference is positive and the difference is negative in case of overlap effect between i and j. The interaction indexes of criteria i and j are defined by Equation 2.

$$I(ij) = \sum_{T \subseteq N \setminus ij} \frac{(n-t-2)!t!}{(n-1)!} \left[\mu(T \cup ij) - \mu(T \cup i) - \mu(T \cup j) + \mu(T)\right] \quad (2)$$

With interaction indexes, a problem involving n criteria will require 2^n coefficients. As the user is not able to specify a such amount of information, we can confine ourself to the 2-order case that permits to model interactions between criteria while remaining simple. Only $n(n+1)/2$ coefficients are then required to define the fuzzy measure. Moreover, in a QoS based selection approach, interactions among more than two quality properties are difficulty interpretable. The coefficients are given by $\mu(i) = a(i)$, the interacting coefficients by $\mu(ij) = a(i) + a(j) + a(ij)$, $i, j \subseteq N$ and the Choquet integral of the utility of alternative x becomes $C_\mu(x) = \sum_{i \in N} a(i)x_i + \sum_{i,j \subseteq N} a(ij)(x_i \wedge x_j), x \in \mathbb{R}^n$.

4.2 Building the reference set and ranking of alternative services

The main step in our selection approach is to derive weights of interacting criteria to apply them to existing service alternatives. These are computed on the basis of a reference set. The reference set is build by the service selector which refers on relationships information provided by the user and the provider QoS models. It consists of fictitious service alternatives and their respective QoS performances ranked with a partial order. Service QoS performances must be expressed on the same scale [7]. Indeed, QoS values are usually stated with different units, according to their respective type or modality. Some QoS Characteristics are defined in percent, others in levels and some in time unit. Moreover, some tend to be minimized while other should be maximized, in accordance with their requester QoS Preferences specification. In the aim to consider all properties on the same scale [9], a preparatory conversion must be made. This conversion consists of:

- **Unifying the unit** The first step is to choose a common unit to all QoS considered. E.g.: the marks will be attributed on a 20 mark or in percent.
- **Setting the modality** All quality attributes must be optimized on the same modality, i.e.: increasing or decreasing. If we choose to maximize all properties, attributes that are usually minimized (e.g.: latency, cost) will inverse their marks. E.g.: a 100 mark for the latency is the quickest latency possible.
- **Scaling of quality attributes** The last element to consider is the scale, all attributes need to be expressed on the same basis. This basis is specified by the unit chosen, properties not directly expressible on this unit must

be transformed. E.g.: if the quality property has a level unit (i.e.: as the security), the transformation function is $actual\ value \times \frac{best\ mark}{max\ value}$. If the property is expressed with a time value like the latency, the transformation function is $1 - \frac{actual\ value - min\ value}{max\ value - min\ value}$ expressed with the chosen unit.

In addition to a partial ranking of service alternatives, the reference set contains specific informations. These informations are detailed here:

- **Importance of criteria** The relative importance of criteria in the service selection approach are compared to the priorities fixed over QoS Characteristics of each service. As these priorities have been established by the user with help of its QoS model, the strength of priorities can be integrated in the reference set with values used. It is also possible to bind these priorities to particular conditions by making them appear in the ranking of service alternatives provided by the service selector in the reference set.
- **Interaction between criteria** This information refers in our selection approach to the dependencies specified by the provider with QoS Dependencies that appear between QoS Characteristics. These appear in the reference set established by the service selector. Theirs strengths and theirs directions can easily be expressed in the initial data of the reference set. Likewise, binded conditions can be included added to the reference set.
- **Symmetric criteria** Symmetric criteria refer to criteria that can be exchanged without changing the aggregation mode. Characteristics belonging to the same QoS Category may sometimes appear as being substitutable, a poor performance in a parameter being compensated by good results in another. Such information needs to be explicitly attached to the parameters of the reference set.

Once all identified relationships among QoS Characteristics appear in the reference set, its corresponding Choquet integrals may be computed thanks to algorithm specified in [15, 16]. Next, to establish Shapley's Value and interaction indexes, linear programming is made on Choquet integrals. Once these weights are fixed, these can be used to determine the performance of services made available by providers. The service selector restricts available services to those that satisfy user functional expectations and constraints on non-functional requirements. Their QoS score need to have been previously scaled as those of reference set alternatives. The ranking of available services is given by the sorting of their respective performance that provides the best available service satisfying user requirements.

4.3 Motivating example

To illustrate clearly contributions and advantages of our approach, we refer to the Example 1 illustrating the utilization of the Priority submodel. Values provided by the service selector to compose the reference set are available on Table 1.

The ranking of alternatives constituent the reference set is given by: $a \succ b \succ c \succ d$. $b \succ c$ and $a \succ d$ are evident preferences. $a \succ b$ and $c \succ d$ are consequences of the priority condition expressed in Example 1 specifying that the availability

Table 1. Example: reference set

Alternative	Availability	Latency	Reliability
a	85	90	90
b	90	85	90
c	90	85	80
d	85	90	80

is more important than the latency if the reliability is under 85%. Similarly, the latency will be favored to availability while reliability is above 85%.

Shapley's values and interaction indexes obtained with help of linear programming on Choquet Integrals associated to the reference set are proposed in Table 2. δ fixes the indifference threshold and has been set to 0.2.

Table 2. Example: Shapley's values and interaction indexes

Quality property	Shapley's value
Availability	0.25
Latency	0.25
Reliability	0.5

	Latency	Reliability
Availability	0	-0.5
Latency	-	0.5

The respective performance of alternatives of the reference set are given in Table 3.

Table 3. Example: Scores of reference set's alternatives and available services and their respective performances

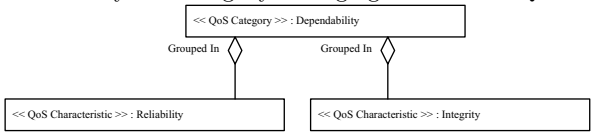
Alternative	Score
a	90.0
b	87.5
c	85.0
d	82.5

Alternative	Availability	Latency	Reliability	Score
e	77	87	92	89.50
f	94	78	85	76.00
g	75	87	91	89.00
h	78	94	87	87.00
i	97	86	78	87.50

Once Shapley's values and interaction indexes are known, these can be used to calculate quality score of providers' alternatives. Table 3 provides quality properties of available services and their respective scores. The best alternative is the service e which proposes a score of 89.50.

The user of the Meris MGVI service will now consider four characteristics rather than three. To this aim, he adds to its specification the QoS Characteristic *integrity* and he specializes this characteristic as belonging to the same QoS Category than the reliability as illustrated in Figure 8. When characteristics are in the same category, these may be substitutable, compensating one characteristic with a poor performance by another with good results.

Fig. 8. Reliability and Integrity belonging to the same QoS Category



The reference set is modified in Table 4 to account for the integrity characteristic. The indifference threshold has been fixed to 0.01 and the interaction threshold to 0.05.

Table 4. Example: modified reference set

Alternative	Availability	Latency	Reliability	Integrity
a	85	90	90	80
b	90	85	90	80
c	90	85	80	80
d	85	90	80	80

The Shapley's value and interaction indexes obtained with the modified reference set are given in Table 5.

Table 5. Example: Modified Shapley's values and interaction indexes

Quality property	Shapley's value
Availability	0.2815
Latency	0.2395
Reliability	0.2395
Integrity	0.2395

	Latency	Reliability	Integrity
Availability	0.05	0.05	0.378998
Latency	-	0.378998	0.05
Reliability		-	-0.05

Table 6. Example: Scores of reference set's alternatives and available services and their respective performances

Alternative	Score
a	85.00
b	83.56
c	80.67
d	80.46

Alternative	Availability	Latency	Reliability	Integrity	Score
e	77	87	92	75	80.69
f	94	78	85	91	84.60
g	75	87	91	86	80.90
h	78	94	87	79	81.91
i	97	86	78	84	82.07

The performance of alternatives of the modified reference set and those of alternatives advertised by providers are given in Table 6. The best available service in response to user's specification is now the service f with a score of 84.60.

This example illustrates the possibilities given by fuzzy MCDA to service selection. The reference set is built on user non-functional requirements and fixed weights reflect its expectations. The obtained ranking of available services is totally lead by relationships identified over QoS Characteristics.

5 Related work

In this section, we overview some existing models in subsection 5.1 and we place our selection approach in existing work in subsection 5.2.

5.1 QoS Models

In attempts to define formally non-functional properties of services, different QoS models have been proposed. Some introduce a description of some concepts [14, 28] while others provide a complete definition of modeling constructs with a linked XML specification [6, 29]. To reach a compromise between a natural conceptualization and an exhaustive formal definition of non-functional characteristics, some authors have used the Unified Modeling Language (UML) to enable QoS modeling [1, 2, 11, 20, 24]. Among them, the Object Management Group (OMG) outlines in [20] a standard, made of UML Profile extensions, to model quality of services. Our definition of two distinct models to, respectively, user and provider and their respective relationships are based on the standard proposed by the OMG.

This standard and most QoS model propositions [14, 28] highlight a clear distinction of QoS characteristics and their quantification. We allow to define quality relationships at characteristics level as at their quantitative calculation level to benefit from this separation.

Our identified relationships over QoS characteristics are identified in some models:

- The preference relationship is introduced in most models with help of a direction attribute [10, 14, 19, 26, 28] indicating if a QoS characteristic has to be maximized or minimized;
- The priority relationship is defined with means of a weight attribute associated to QoS characteristics [14, 28];
- The dependency relationship has only been summarily addressed in some QoS models [17, 20] without specific attribute.

However, none of the cited models provides QoS constructs needed to account for all of the considerations defined in our QoS models. We have shown the relevance of these considerations, as they are needed in comparing competing services and subsequently selecting the most appropriate ones. Although some of the cited models define constructs, some of which are intended for service providers, others for service requesters, we separate the two perspectives and thereby make a clear and explicit distinction between the models of the two parties.

5.2 QoS Driven Selection

The aim of service selection is to affect the most suitable service to each service request. If the selection is based on non-functional considerations of services, it will result on a matching of providers QoS capabilities and requesters QoS requirements. Different techniques have been proposed to process this match, among them: multi-criteria decision making [19, 25, 28]; fuzzy MCDA [26]; heuristics [10]; Euclidean distances [14] and; reputation models [17]. Some propositions combine several techniques as Vu et al. [27] whose use data-mining with a reputation model and multi-criteria decision making. Some of these propositions are integrated in larger approaches that compose Web services [8, 10, 28] while others are confined to the definition of the most adapted service to user requirements.

Existing service selection approaches do not account for relationships over QoS Characteristics. However, the priority relationship is threated by means of weighting in some models [25, 27, 28]. These weights allow to indicate the relative importance of each QoS Characteristic considered during the selection step [8]. Our proposed selection approach relying on fuzzy MCDA allow for the integration of all identified relationships over QoS Characteristics. This way, the ranking established with our technique benefits from advanced concepts specified by providers and requesters.

6 Conclusions and Future Work

In a Service-Oriented System (SOS), service requesters specify tasks that need to be executed and the quality levels to meet, whereas service providers advertise

their services' capabilities and the quality levels they can reach. Selecting appropriate services among competing ones requires rich QoS models for describing services' QoS dimensions and characteristics, as this information is subsequently needed to inform comparison and decision-making during selection. We introduce an approach that features rich QoS models and a service selection method that uses the QoS information available in the models. The approach consists of: (i) rich QoS models to be used by service requesters when expressing QoS expectations and service providers when describing services' QoS, and for representing preference and priority relationships between QoS dimensions; and (ii) a multi-criteria decision making technique that uses the models for service selection. The approach therefore allows us to deal with tradeoffs through priorities, and to account for stakeholders' preferences over values of QoS dimensions and characteristics.

Future effort will be focused on the automation of the approach. Namely, automated matching of provider and requester QoS model instances will be proposed in addition to an algorithm for building the reference set while accounting for existing QoS relationships.

References

1. J. O. Aagedal, and E. F. Ecklund Jr.. Modelling QoS: Towards a UML Profile In *Proceedings of the fifth International Conference on the Unified Modeling Language (UML'02)*, 275–289, 2002.
2. J. I. Asensio, V. A. Villagra, J. E. Lopez de Vergana and J. J. Berrocal. UML Profiles for the Specification and Instrumentation of QoS Management Information In Distributed Object-based Applications. In *Proceedings of the fifth world multiconference on systemics, cybernetics and informatics*, 22–25, 2002.
3. B. Benetallah and F. Casati. Special Issue on Web Services. In*Distributed and Parallel Databases*, 12, 115–116, 2002.
4. F. Casati, M. Castellanos, U. Dayal and M. C. Shan. Probabilistic, context-sensitive, and goal-oriented services selection. *ICSOC '04: Proceedings of the 2nd international conference on Service oriented computing*, 2004.
5. L. Chung, B. A. Nixon, E. Yu and J. Mylopoulos. Non-Functional Requirements in Software Engineering (The Kluwer International Series in Software Engineering Volume 5), Springer, 1999.
6. A. D'Ambrogio. A model-driven WSDL Extension for Describing the QoS of Web Services. In *Proceedings of the International Conference on Web Services (ICWS'06)*, 2006.
7. J. Figueira, S. Greco and M. Ehrgott. Multiple Criteria Decision Analysis: State of the Art Surveys. Springer Verlag, 2005.
8. X. Gu and K. Nahrstedt. A Scalable QoS-Aware Service Aggregation Model for Peer-to-Peer Computing Grids. *HPDC '02: Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing*, 2002.
9. C. L. Hwang, K. Yoon. Multi-Attribute Decision Making: Methods and Applications. Springer-Verlag. 1981.
10. M. C. Jaeger, G Rojec-Goldmann and G. Mühl. QoS Aggregation for Web Service Composition using Workflow Patterns. *EDOC '04: Proceedings of the Enterprise Distributed Object Computing Conference, Eighth IEEE International*, 2004.

11. I. J. Jureta, C. Herssens, S. Faulkner. A Comprehensive Quality Model for Service-Oriented Systems. *Software Quality Journal*. Accepted for publication (available online at: http://www.jureta.net/papers/QVDPdraft.pdf).
12. S. Kalepu, S. Krishnaswamy and S. W. Loke. Verity: A QoS Metric for Selecting Web Services and Providers. *WISEW'03: Proceedings of the fourth International Conference on Web Information Systems Engineering Workshops*, 2003.
13. A. Keller and H. Ludwig. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. *Journal of Network Systems Management*, 11(1), 2003.
14. Y. Liu, A. H. Ngu and L. Z. Zeng. QoS computation and policing in dynamic web services selection. *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, 2004.
15. J.-L. Marichal and M. Roubens. Determination of weights of interacting criteria from a reference set. *European Journal of Operational Research*, 124, 641–650, 2000.
16. J.-L. Marichal. Aggregation of interacting criteria by means of the discrete Choquet integral. In *Studies in Fuzziness and Soft. Computing*, 97: 224-244, 2002.
17. E. M. Maximilien and M. P. Singh. Toward autonomic services trust and selection. *ICSOC'04: Proceedings of the International Conference on Service-Oriented Computing*, 2004.
18. D. A. Menascé. QoS Issues in Web Services. *IEEE Internet Computing*, 6(6), 72–75, 2002.
19. F. Naumann, J. C. Freytag and U. Leser. Quality-driven Integration of Heterogeneous Information Systems. Proceedings of th 25th VLDB Conference, 1999.
20. The Object Management Group. UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms. *Adopted Specification*, 2006.
21. J. O'Sullivan, D.Edmond and A. Ter Hofstede. What's in a Service? Towards accurate description of non-functional service properties. *Distrib. Parallel Databases*, 12(2-3), 117–133, 2002.
22. M. P. Papazoglou and D. Georgakopoulos. Service-Oriented Computing. In *Communications of the ACM*, 46(10), 25–28, 2003.
23. S. Ran. A Model for Web Services Discovery with QoS. *ACM Sigecom exchanges*, 2003.
24. G. Salazar-Zarate, P. Botella. Use of UML for modeling non-functional aspects. In *Proceedings of the International Conference on Software and Systems Engineering and their Applications (ICSSEA00)*, 2000.
25. S. E. Shaikh and N. Mehandjiev. Multi-Attribute Negotiation in E-Business Process Composition. *WETICE '04: Proceedings of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2004.
26. H. Tong and S. Zhang. A Fuzzy Multi-attribute Decision Making Algorithm for Web Services Selection Based on QoS. *APSCC '06: Proceedings of the IEEE Asia-Pacific Conference on Services Computing*, 2006.
27. L.-H. Vu, M. Hauswirth and K. Aberer QoS-Based Service Selection and Ranking with Trust and Reputation Management. in*Proceedings of the 13th International Conference On Cooperative Information Systems (CoopIS05)*, 2005.
28. L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam and H. Chang. QoS-Aware Middleware for Web services composition. *IEEE Trans. Softw. Eng.*, 30(5), 311–327, 2004.
29. C. Zhou, L.-T. Chia and B.-S. Lee. Daml-qos ontology for web services. *ICWS'04: Proceedings of the International Conference on Web Services*, 2004.