Enhancing Service Selection by Semantic QoS

Henar Muñoz Frutos¹, Ioannis Kotsiopoulos², Luis Miguel Vaquero Gonzalez¹,
and Luis Rodero Merino¹

¹ Telefónica Research & Development, C/ Emilio Vargas, 6
28043 Madrid, Spain

² School of Computer Science, University of Manchester, Kilburn Building
Oxford Road, Manchester, M13 9PL, United Kingdom
henar@tid.es, ioannis@cs.man.ac.uk, lmvvg@tid.es, rodero@tid.es

Abstract. The increasing number of functionally similar services requires the existence of a non-functional properties selection process based on the Quality of Service (QoS). Thus, in this article, authors focus on the provision of a QoS model, an architecture and an implementation which enhance the selection process by the annotation of Service Level Agreement (SLA) templates with semantic QoS metrics. This QoS model is composed by a specification for annotating SLA templates files, a QoS conceptual model formed as a QoS ontology and selection algorithm. This approach, which is backward compatible, provides interoperability among customer-providers and a lightweight alternative. Finally, its applicability and benefits are shown by using examples of Infrastructure services.

Keywords: Service Selection, QoS, semantic annotations, SA-SLA, ontology.

1 Introduction

The marketplace model, where customers and providers interact for buying and selling services, is evolving to incorporate infrastructure services such as storage or processing ones [1], which can be accessible through the Internet [2] boosting the number of available services [3]. In this setting of increasing number of functionally similar services, the discovery process to meet user requirements while ensuring the QoS [32] can be seriously hampered. Therefore, there is a need for a service selection process based on specified customer preferences or restrictions which are considered as non-functional properties [31] [32] such as price, reputation and reliability. Some of the non-functional properties can involve QoS attributes which may impact the quality of the service offered by a Web Service [15] [32]. Moreover, this QoS specification can be contained inside the Service Level Agreement (SLA) Templates (TSLA) files which are used to achieve an agreement between customer and provider about the terms required in the service provision [18][20]. Thus, a service specification offered by a provider can be materialized as a capability (functional description) provided in the service description and a set of QoS constraints (non-functional descriptions) in the associated TSLA files.

In this highly dynamic scenario, providers and customers may use different terminologies or even different languages, to describe the conditions and requisites [21] (like *Performance*, *Throughput* or *Response Time*). This heterogeneity in data and interfaces may turn the selection process almost impossible when using the conventional approaches. For instance, UDDIe [16] and the QoS Broker developed by Serhani et al [17] extend the Universal Description, Discovery and Integration (UDDI) structure to include QoS attributes in the service description. However, they model QoS metrics syntactically, which is poor and not extensible [15], and cannot solve the heterogeneity problem. In this matter, it is required a mutual agreement on the service(s) functional and non-functional properties [21] [22] [23] to develop a method for automatically matching the offers and the requests [22].

In order to increase the interoperability among customers-providers, some works introduce semantic technologies in the definition of QoS metrics. Wang et al [27], Tsesmetzis et al [30], Maximilien et al [33] and Ren et al [32] present different conceptual models formed as ontologies so as to extend service descriptions. However, these solutions are not backward compatible, and demand providers to change their service offer descriptions. In addition, Oldham et al [22] and Chaari et al [23] propose the definition of QoS metrics inside standards to manage non-functional aspects of WS-Agreement [18] and WS-Policy [19] respectively, so that, they do not modify current service descriptions. Although, they introduce semantic annotations inside the standards, they extend their standards XML schemas, being alternatives not backward compatible.

Thus, in this paper we try to tackle the detected inflexibility of current QoS metrics specification approaches providing a backward compatible approach. We enrich the expression capabilities of current QoS frameworks by developing an agreed conceptual model captured in the form of ontologies for improving the interoperability among parties and a semantic specification for annotating SLA files. Thus, the contribution is related to the provision of a QoS model composed by i) a semantic specification for annotating SLA files and ii) a QoS ontology which contains also non-functional properties for Infrastructure services. Moreover, a reference architecture is designed to implement this QoS model and an implementation to test its applicability and benefits.

The remaining of this document is structured as follows. Section 2 presents the related work; Section 3 shows the proposed QoS model including the semantic annotations and the ontology. This framework is mapped to a reference architecture (Section 4) and validated with experiments in Section 5. Finally, Section 6 summarizes the obtained results and presents the future work.

2 Related Work

Work in the area of Grid/Web Service (GWS) QoS typically involves syntactic aspects. Serhani et al [17] presents an architecture based on QoS certification approach employing an extended UDDI for supporting Web Service selection using QoS attributes. Moreover, UDDIe [16] extends the structure to include QoS attributes in the context of Grid computing. In addition, other initiatives enable standardized QoS specification for Web Service defined on top of existing protocols and standards for the establishment of an agreement between two parties. There are two main standards related to SLA, which use QoS metrics to define the terms agreed: i) Web Services

Agreement Specification (WS-Agreement) [18], which extends current Web Services standards to allow the establishment of an agreement and ii) Web Service Level Agreement Language (WSLA) [20] which provides a framework for monitoring and evaluating SLA. Moreover, some work has been carried out towards the combination of both standards with the result of the TrustCoM SLA specification [21]. Moreover, before signing the contract, TSLA files are used to negotiate the QoS metrics [21] in order to achieve an agreement between customer and provider about the terms required in the service provision. Thus, the provider describes the service offers in terms of QoS metrics in the TSLA.

In spite of helping guaranteeing QoS levels, these initiatives do not support open environments [15], where flexible QoS is a must to enhance the interaction and understanding between providers and consumers. The reason is because GWS definition languages are mainly XML-based [4], which lacks the semantic expression power for its flexible definition [7] [8] in an evolving market place where many consumers and providers often express their preferences using different terminologies. In fact, machines are not able to process some information, which is only comprehensible by humans [9]. As a result, services are rigid and cannot be adapted to changes without human intervention [7]. Ontologies can overcome the heterogeneity problem providing some automatic support [9], since they allow data semantics to be expressed in a formal and more expressive language understandable by machines [7].

Similar to our approach, other works in the area of GWS QoS employ semantic technologies to increase interoperability. Wang et al [27], Tsesmetzis et al [30], Maximilien et al [33] and Ren et al [32] present different models for discovery services with QoS constraints by using ontologies as well as some algorithms for matchmaking purposes. However, as they extend current service descriptions, these solutions are not backward compatible, and demand providers to change their service offer descriptions. Moreover, the entire approach can lack in efficiency and performance, since all customers' requests are translated into rules and ontologies increasing the time in providing the answer. On the other hand, Tian et al [24] offer a specification of several QoS aspects and a mapping of QoS requirements regarding network performance onto the actual transport technology at runtime. However, this work is focused towards network parameters monitoring and not business requirements.

Finally, the approach discussed here is similar to the contribution presented by Oldham et al [22], which enriches WS-Agreement with semantic annotations. This approach specifies the QoS metrics inside the standards instead of the service description. However, the solution provided is limited due to:

- It annotates TSLA by using semantic QoS metrics extending the current WS-Agreement schema. This approach fails since it is not backward compatible, so that, previous components, which do not understand the new specification, cannot work with it.
- It transforms all customer requests into ontology and rule descriptions, which makes the process be so slowly, since it implies an updating of the ontology each time a request is done.

In order to improve interoperability between customer and provider and have a backward compatible solution, which allows for working in open environments, our approach provides a QoS model, which is explained in next Section.

3 Selection of Services: The QoS Model

This work proposes a QoS model to allow for service selection based on QoS metrics, which are specified inside TSLA files, where providers describe their offers in terms of non-functional properties. Moreover, in order to increase the interoperability among customers and providers, we are going to introduce semantic technologies in terms of semantic annotations of the QoS metrics included in the TSLA. The main drivers behind this approach are: i) the necessity for a backward compatible approach, so that, components non aware of semantic annotation can continue working with SLA files (there will be annotated and not-annotated SLA files) and ii) a lightweight approach to perform any operation (selection, negotiation...) at runtime [21]. In order to implement this approach, a QoS model is defined, which is composed by i) a specification for introducing semantic QoS annotations in SLA files, ii) a conceptual model for QoS metrics formed as ontologies to be shared between customers and providers and iii) a QoS selection algorithm. For the selection algorithm, we took the work done in Wang et al [27], since it tries to solve similar problems to ours. In this work, we skip QoS metrics collection as part of the QoS model, since we assume that providers have already obtained and measured the QoS metrics values to be included in the TSLA files.

Fig. 1. A QoS Model example

Thus, in order to analyze the QoS model, we use an example obtained from the Virtual Scenario of the BREIN project [34] which is represented in Figure 1. There, an engineering company (ANSYS) tries to outsource the computing infrastructure required for software simulations in order to reduce costs. In this matter, due to the existence of several infrastructure providers (BSC, HLRS...), the framework should be able to select the best provider in each moment which satisfies the customer's requests. In the example, the customer asks for a computing service together with some QoS parameters like "memory higher than 6" and "response time lower than 10" to

guarantee the service to the final clients. In real business, clients and providers can use different terminologies, or even different languages, to describe their requests or offers. This situation is shown in the example, where the customer and the providers are using different terminologies and even languages (e.g. *Response time* vs. *Throughput* vs. *Rendimiento* (*Performance*)). Thus, the selection process cannot return any answer in case a keyword search is carried out.

In addition, Figure 1 represents the previous example in the specification of the QoS model. By using the QoS model, the SLA parameters in the TSLA files are annotated by QoS terms which are linked to concepts from the QoS ontology, extending the common conceptual model. In addition, the TSLA files obtain a score indicating the customer' satisfaction (in term of customer' conditions) by the QoS selection algorithm.

3.1 Specification of Semantic Annotations for SLA

Our work provides a specification for annotating TSLA files called Semantic Annotation for Service Level Agreement (SA-SLA). SA-SLA is based on the Semantic Annotations for Web Service Description Language (SA-WSDL) [5][6], which has become the dominant approach in the area of Semantic Web Services [20]. It provides a standard description format extending SLA template files such as WS-Agreement, TrustCoM templates or WSLA. Thus, it allows for annotating SLA descriptions with pointers to semantic concepts from more expressive ontologies, coded in formal languages as Web Ontology Language (OWL) [11] or Web Service Modelling Language (WSML) [13], so that SLA-aware components can interpret the content and automate the tasks. Concretely, in this paper contribution, we are linking SLAParameters and Metrics in TSLA with semantic descriptions by using the QoS concepts ontology.

Fig. 2. TSLA example annotated with SA-SLA specification

The SA-SLA specification is constituted mainly by: i) model reference, which is an association between a SLA schema and a concept in some semantic model, and ii) schema mappings for specifying mappings between semantic data and XML [6]. Figure 2 represents an example of how an SLA template according to TrustCoM SLA specification can be annotated by some concepts (*Performance* and *ResponseTime*) from the BREIN QoS ontology following the SA-SLA specification, mainly by using the model reference element.

3.2 QoS Ontology

Since services can be provided by third parties and invoked dynamically over the Internet, their QoS can vary greatly. Therefore, it is important to have a framework capturing the QoS provided by the supplier and for the QoS required by the customer, and ultimately the match between the two when discovering the service best match the required QoS [29].

Fig. 3. QoS ontology

Thus, BREIN provides a QoS conceptual model, which is shared by customers and providers, and formed as an OWL QoS ontology to provide a common understanding of QoS parameters and their semantics. The ontology consists of basic concepts to define QoS parameters and the relationship between them required for describing non-functional properties for services included Infrastructure ones. Basically, the QoS

ontology is based on the QoS conceptual model provided by Wang et al [27], Lee et al. [28], Shuping et al [29], Tsesmetzis et al [30], Maximilien et al [33] and Ren et al. [32], and it will be extended in the future work to relate business QoS metrics like availability to infrastructure ones.

The QoS ontology defines a generic set of QoS parameters, which is extensible, and can be customized to include domain-specific QoS parameters, as Figure 3 depicts. In order to conceptualize this domain, we organize them into QoS categories, which are grouped into different types: i) system, which describes non-functional properties in systems which are encapsulated as services, ii) network, which involves the typical QoS requirements in the network, iii) infrastructure, related to requirements coming back infrastructure services or task to be executed in the resources, and iv) some extension concepts focused on security and economical issues.

Moreover, according to the example shown in Figure 1, a method to work with different terminologies of the same concept in the ontology is required (e. g. “Response Time” vs. “Tiempo de Respuesta” with respect to qos:ResponseTime concept). Thus, the QoS model uses the Simple Knowledge Organization System (SKOS) [10], which provides a standard way to represent knowledge organisation systems using the Resource Description Framework (RDF) [14], so that, it allows for linking different representations to a QoS concept. In this matter, concepts in the QoS ontology can be mapped into different representations by SKOS files. One example of this specification can be seen on Figure 4, where the QoS concept ResponseTime is mapped into several different representations corresponding to different languages.

```
<rdf:Description rdf:about="http:// brein.com/Upper/QoS#ResponseTime">
<skos:inScheme rdf:resource="http://eu-brein.com/thesaurus/QoSmetrics"/>
  <rdf:type
rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
    <skos:prefLabel xml:lang="en">Response Time</skos:prefLabel>
    <skos:altLabel xml:lang="es">Tiempo de Respuesta</skos:altLabel>
  /rdf:Description>
```

Fig. 4. SKOS representation example

4 Implementation

The previous QoS model needs to have a framework which implements its specification, conceptual model and selection algorithm. Figure 5 shows an architecture, whose main two modules have been implemented inside the BREIN project. Mainly, it is composed by:

- **Semantic Enhanced Service Selector (SESS):** This component allows for parsing of TSLA annotated files offered by providers and matched the providers' offers with the customer' parameters. It will interact with the SLA Translator (SLAT) which supports SESS to understand the semantic annotations. Moreover, the selector module implements the algorithm proposed. As a result, the best service, which satisfies customer's request, is returned.
- **SLA translator:** It provides the semantic support for SLA aware components, concretely by the translation of providers' metrics into predefined

ones, solving the heterogeneity problem commented above. It is composed by an ontology manager and query engine to manage ontologies. Moreover, it has an SKOS module to take into account different representations of the same concept through the SKOS standard.

Fig. 5. The framework which implements the QoS model

These two main components are composed by a set of functional modules:

- **TSLA Parser:** It parses the TSLA files, which define the providers' offers, in order to obtain the QoS metrics required by the customer and their values. Thus, it performs two kind of matchmakings depending on the metrics (annotated or not): i) syntactic, which involves a keyword search and ii) semantic, which interact with SLAT in order to matchmake the semantic QoS metrics with customer' requests.
- **Service Provider (SP) SLA repository:** It stores the TSLA templates associated to the different services candidates in the selection process.
- **TSLA Selector:** It applies a selection algorithm taken from the work done by Wang et al [27]. This algorithm normalizes the matrix constituted by QoS metrics (Q_s) to map all variables to a common range [0, 1] by scaling the value ranges with the maximum and minimum values of each quality metric [27]. Finally, it sums the normalized metrics value in order to obtain the TSLA score. Thus, the selector supplies each provider's score and chooses the one which has the highest score.
- **Ontology Manager:** It allows for managing the QoS ontology, by using Jena as ontology manager and Pellet as reasoner working with OWL ontologies.
- **SKOS Manager:** It allows for managing the SKOS representation of the QoS concepts. It also uses Jena to work with RDF representations.

5 Experimentation

The scenario chosen for demonstration has been explained in Section 3 and shown in Figure 1. Thus, for experimentation proposes, we built an early simulation testbed composed by a set of TSLA files (6 files concretely) based on the TrustCoM specification, which are associated to different services with similar functionality (processing capacity), so that, a selection process is required. These TSLA files have different QoS metrics to characterize the service offers. We are supposing the usage of different terminologies in the metric definition (as performance vs. response time or memory vs. storage capacity) as well as different languages (English, Spanish, and French), which is the problem found in the open environments. Moreover, some TSLA are semantically annotated by QoS concepts from the QoS ontology. Figure 6 represents this testbed, which includes a customer' request asking for 4 QoS metrics and different providers' TSLA described by using some of these metrics, which can be semantically annotated.

Fig. 6. SLA templates testbed

Figure 7 shows a representation of the results obtained by testing the implementation of the QoS model. Concretely, it presents the total score of each TSLA which is formed as the sum of the normalized value of the QoS metrics in the TSLA involved in the testbed. As a result of the experimentation, we see that service #1 although having all metrics required by the customer and the best values in the testbed, its score is 0, since it lacks of semantic annotations and the terminology used in the metrics definition is different from the customer one. Thus, as the selector module is doing a syntactic search (due to the lack of semantic annotations), there are not results in the matchmaking process. On the other hand, despite the fact that service #6 does not satisfy all customer' restrictions and their values are worst than service #1, it has the best score, since all its metrics have been annotated.

In addition, we can observe that service #5, in spite of using different annotations than the customer (e.g. qos:ResponseTime vs. qos:Performance or qos:Cost vs. qos:Price), its metrics are matchmade, since they are using related concepts belonging to QoS ontology. Furthermore, we demonstrate that it is possible to work with non-annotated metrics, only if they are defined by the customer' terminology as the Availability metric in service #3.

Fig. 7. Experimentation results from the implementation (where MEM= Memory User For Task, RES=Response Time, AVA= Availability, COST=Price)

However, we see that the TSLA scores obtained do not fully satisfy the customer' request. For instance, there is not any penalization in case the TSLA does not include metrics asked by the customer. Service #4 has the third position in the ranking despite the fact it does not consider two metrics (Response Time and Availability). On the other hand, service #2 considers most metrics annotated and its score is low. In fact, due to the way of scaling by maximum values, the metric Response Time in service #2 although it is so close to the customer one (10.1 vs. 10), for being a maximum, its normalized value is 0. Finally, although a normalization in metrics values is carried out by the selection algorithm [27] to map all variables to a common range [0,1], the output of this result is not scaled into [0,1].

Thus, these results show that in an open market, where new customers and providers can enter, mechanisms to improve several aspects such as interoperability, performance and easy of use are key aspects. In this matter, improving semantic interoperability during the selection phase implies that more providers are able to be introduced in the market. Moreover, by allowing the users to define their own terms for TSLA descriptions and reducing the task of selection process from customer's requirements by automatic support, we achieve decreasing the entrance barrier of new actors in the market. However, the algorithm used is not the best one due to the limitation analysed by the examples.

6 Conclusions and Future Work

The increasing number of functionally similar services requires the existence of a non-functional properties selection process. This work has focused on the service selection based on QoS metrics used inside TSLA files. Moreover, it has introduced

semantic technologies for annotating QoS metrics to increase the interoperability between customers and providers, which can use their own and different terminologies. Thus, our contribution has provided a QoS model composed by i) SA-SLA specification and ii) a QoS ontology. Mainly, the formalization of the QoS conceptualization model in terms of ontology increases the interoperability as the experimentation has shown. In addition, the SA-SLA specification makes the QoS model be backward compatible and a lightweight approach, which allow for continuing working with previous implementations. Furthermore, our work provides a reference architecture together its development which implements the QoS model proposed. From our experimentation, it is possible to see the advantages of using semantics in this open environment.

As future work, we will make the QoS ontology more expressive and improve the relationships between QoS metrics mapping the business requirements to infrastructure ones. Moreover, the QoS ontology will be part of a conceptualization model for business Grid, in order to include more non-functional properties in the selection process (also related to the resources which the Infrastructure services virtualize). In addition, we plan to continue working in the QoS model in order to extend the algorithm we took to adjust/extend it to our requirements considering the feedback we obtained from the experimentation. Basically, the idea is to change the normalization method for scaling metrics, to include penalization in case not all the customer's conditions have included and to modify the algorithm so that output is scaled also between $[0,1]$. Finally, current implementation will be extended to take into account previous changes. Moreover, we will try to apply some Peer To Peer technologies in order to provide scalability to our solution required when the number of entities improves.

Acknowledgments

This work has been supported by the BREIN project and has been partly funded by the European Commission's IST activity of the 6th Framework Programme. This paper expresses the opinions of the authors and not necessarily those of the European Commission. The European Commission is not liable for any use that may be made of the information contained in this paper.

References

1. Brunner, R., Freitag, F., Navarro, L.: Towards the development of a decentralized market information system: Requirements and architecture. In: IEEE International Symposium on Parallel and Distributed Processing IPDPS 2008, pp. 1–7 (2008)
2. Shaw, M.J.: Building an E-Business from Enterprise Systems. *Information Systems Frontiers* 2(1), 7–17 (2000)
3. Cusumano, M.A.: The changing software business: Moving from products to services. *Computer* 41(1), 20–27 (2008)
4. Papazoglou, M.P., Heuvel, W.J.: Service oriented architectures: approaches, technologies and research issues. *The VLDB Journal* 16(3), 389–415 (2007)

5. Kopecký, J., Vitvar, T., Bournez, C., Farrell, J.: SAWSDL: Semantic Annotations for WSDL and XML Schema. *IEEE Internet Computing* 11(6), 60–67 (2007)
6. Farrell, J., Lausen, H.: Semantic Annotations for WSDL and XML Schema, W3C. Recommendation (August 28, 2007), <http://www.w3.org/TR/sawSDL/>
7. Gruber, T.R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In: Guarino, N., Poli, R. (eds.) *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Dordrecht, The Netherlands. Kluwer Academic Publishers, Dordrecht (1993)
8. Fensel, D., Lausen, H., Polleres, A., de Bruijn, J., Stollberg, M., Roman, D., Domingue, J.: *Enabling Semantic Web Services: The Web Service Modeling Ontology*. Springer, New York (2006)
9. Krummenacher, R., Hepp, M., Polleres, A., Bussler, C., Fensel, D.: Wwww or what is wrong with Web Services. In: *ECOWS 2005: Proceedings of the Third European Conference on Web Services*, Washington, DC, USA. IEEE Computer Society, Los Alamitos (2005)
10. The OWL Services Coalition. OWL-S 1.1 beta release (July 2004), <http://www.daim.org/services/owl-s/>
11. OWL Working Group, Web Ontology Language (OWL) (2004), <http://www.w3.org/2004/OWL/>
12. W3C Semantic Web Deployment Working Group, SKOS Simple Knowledge Organization System Reference, <http://www.w3.org/TR/skos-reference>
13. ESSI WSMML working group, Web Service Modeling Language, <http://www.wsmo.org/wsmml/>
14. RDF Core Working Group, Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C recommendation, <http://www.w3.org/TR/rdf-concepts>
15. Kritikos, K., Plexousakis, D.: A semantic qos-based Web Service discovery algorithm for over-constrained demands. In: *NWESP 2007: Proceedings of the Third International Conference on Next Generation Web Services Practices*, Washington, DC, USA, pp. 49–54. IEEE Computer Society, Los Alamitos (2007)
16. ShaikhAli, A., Rana, O.F., Al-Ali, R., Walker, D.W.: UDDIe: An Extended Registry for Web Services. In: *Proceedings of the 2003 Symposium on Applications and the internet Workshops (Saint 2003 Workshops)*, January 27 - 31, 2003, p. 85. IEEE Computer Society, Washington (2003)
17. Serhani, M.A., Dssouli, R., Hafid, A., Sahraoui, H.: A QoS Broker Based Architecture for Efficient Web Services Selection. In: *Proceedings of the IEEE international Conference on Web Services, ICWS*, July 11 - 15, 2005, pp. 113–120. IEEE Computer Society, Washington (2005)
18. Grid Resource Allocation Agreement Protocol WG (GRAAP-WG), Web Services Agreement Specification (WS-Agreement), Version 2005/09 (September 2005), http://www.ggf.org/Public_Comment_Docs/Documents/Oct-2005/WS-AgreementSpecificationDraft050920.pdf
19. W3C, Web Service Policy, <http://www.w3.org/TR/ws-policy/>
20. Web Service Level Agreement (WSLA) Language Specification, ver. 1.0 (January 2003), <http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf>
21. Kotsiopoulos, I., Soler Juber, S., Tenschert, A., Benedicto Cirujeda, J., Koller, B.: Using Semantic technologies to improve Negotiation of Service Level Agreements, *eChallenges* 2008 (2008)

22. Oldham, N., Verma, K., Sheth, A., Hakimpour, F.: Semantic WS-agreement partner selection. In: Proceedings of the 15th international Conference on World Wide Web, WWW 2006, Edinburgh, Scotland, May 23-26, 2006. ACM, New York (2006)
23. Chaari, S., Badr, Y., Biennier, F.: Enhancing Web Service selection by QoS-based ontology and WS-policy. In: Proceedings of the 2008 ACM Symposium on Applied Computing, SAC 2008, Fortaleza, Ceara, Brazil, March 16 - 20, 2008, pp. 2426–2431. ACM, New York (2008)
24. Tian, M., Gramm, A., Ritter, H., Schiller, J.: Efficient Selection and Monitoring of QoS-Aware Web Services with the WS-QoS Framework. In: Proceedings of the 2004 IEEE/WIC/ACM international Conference on Web intelligence, Web Intelligence, September 20 - 24, 2004, pp. 152–158. IEEE Computer Society, Washington (2004)
25. Somasundaram, T.S., Balachandar, R.A., Kandasamy, V., Buyya, R., Raman, R., Mohanram, N., Varun, S.: Semantic Description and Discovery of Grid Services using WSDL-S and QoS based Matchmaking Algorithm. In: Advanced Computing and Communications, 2006. ADCOM 2006, December 20-23, pp. 113–116 (2006)
26. Li, H., Du, X., Tian, X.: Towards Semantic Web Services Discovery with QoS Support Using Specific Ontologies. In: Proceedings of the Third international Conference on Semantics, Knowledge and Grid, SKG, October 29 - 31, 2007, pp. 358–361. IEEE Computer Society, Washington (2007)
27. 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)
28. Lee, K., Jeon, J., Lee, W., Jeong, S., Park, S.: QoS for Web Services: Requirements and Possible Approaches, W3C Working Group Note 25 (2003), <http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/>
29. Ran, S.: A Model for Web Services Discovery With QoS. ACM SIGecom Exchanges 4(1) (March 2003)
30. Tsesmetzis, D.T., Roussaki, I.G., Papaioannou, I.V., Anagnostou, M.E.: QoS awareness support in Web-Service semantics. In: Proceedings of the Advanced int’L Conference on Telecommunications and int’L Conference on internet and Web Applications and Services, AICT-ICIW, February 19 - 25, 2006, p. 128. IEEE Computer Society, Washington (2006)
31. Garcia, D.Z., de Toledo, M.B.: Semantics-enriched QoS policies for Web Service interactions. In: Proceedings of the 12th Brazilian Symposium on Multimedia and the Web, WebMedia 2006, Natal, Rio Grande do Norte, Brazil, November 19 - 22, 2006, vol. 192, pp. 35–44. ACM, New York (2006)
32. Ren, K., Chen, J., Chen, T., Song, J., Xiao, N.: Grid-Based Semantic Web Service Discovery Model with QoS Constraints. In: Third International Conference on Semantics, Knowledge and Grid, October 29-31, 2007, pp. 479–482 (2007)
33. Maximilien, E.M., Singh, M.P.: A Framework and Ontology for Dynamic Web Services Selection. IEEE Internet Computing 8(5), 84–93 (2004)
34. The BREIN Project, Web site (December 9, 2008), <http://www.eu-brein.com/>