ADDING MEANING TO QOS NEGOTIATION

Cláudia M. F. A. Ribeiro
Universidade do Estado do Rio Grande do Norte, Av. Itapetinga, 1430, Natal, Brazil

Nelson S. Rosa, Paulo R. F.Cunha
Universidade Federal de Pernambuco, Cidade Universitária, Recife, Brazil

Keywords: QoS Negotiation, Semantic Web Services, SLA Ontology.

Abstract: Using quality of service (QoS) to discover Web Services that better meet users' needs became a key factor to differentiate similar services. Treating QoS includes negotiating QoS capabilities, since there is a potential conflict between service provider and service requestor requirements. This paper addresses this problem by using an ontological approach for QoS negotiation that aims to improve user participation. For this purpose, a Service Level Agreement (SLA) ontology that explicitly considers subjective user QoS specification was conceived. Its internal components and the role it plays during service discovery are detailed.

1 INTRODUCTION

Evidence suggests that, by focusing on quality, companies can increase market share. In Web Service market, similar services can be differentiated by quality of service (QoS) properties, which is a set of non functional properties that fully characterizes "how" a service works, instead of "what" it does (Chung, 2000). In multimedia applications, QoS providing means to meet user needs covering more than just price, i.e. other requirements, such as usability and security, can also be used to identify the desired service.

Treating QoS includes negotiating QoS capabilities, since there is potentially conflict between service provider and service requestor requirements. Indeed, the service provider aims to minimize resource allocation, whereas the desired QoS level defined by user tends to increase resource usage. While QoS negotiation requires QoS specification, it is apparent the necessity to include QoS information in Web Services description.

Today's Web Services description is mainly automation oriented, based on application to application communication. It means that service description is focused on how to make service discovery, selection, invocation and monitoring, more automatic processes. For instance, automatic discovery involves automatically locating Web Services that provide a particular service with appropriate properties (Booth, 2005).

Semantic Web Services are considered the next generation of Web Services (Alesso, 2004). They differ from traditional ones by using ontology, such as OWL-S (Martin, 2003) and WSMO (Feier, 2005), to add meaning to their description, although they do not prioritize QoS. Considering that human beings have used more and more Internet as a preferred environment, both for document search and for localizing services, it is essential to consider their QoS needs, since they can perceive changes in QoS level actually provided.

In this work, it is presented a model for semantic QoS negotiation, which is defined as the process of Web Services selection, considering service QoS capabilities. It also includes abstract considerations about resource negotiation for QoS guarantees based on ontology. For this purpose a domain ontology for Service Level Agreement (SLA) was conceived. SLA is also known as QoS contract and may include QoS specification from different perspectives, from subjective user level to low level network mechanisms.

Despite the use of ontologies for QoS, and specifically for SLA were subject of important research activities (Zhou, 2005), (Menascé, 2002), there is a gap in the explicit treatment of human needs during QoS negotiation. A premise used by

M. F. A. Ribeiro C., S. Rosa N. and R. F.Cunha P. (2006).
ADDING MEANING TO QOS NEGOTIATION.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - SAIC*, pages 106-111
DOI: 10.5220/0002463401060111

this work is that meaningful QoS contracts (Ribeiro, 2004a) can be used to improve the user participation during QoS negotiation and monitoring, contributing to user satisfaction. The internal components of SLA ontology proposal and its role at QoS negotiation are described.

This paper is structured as follows. Section 2 is dedicated to related work. Section 3 describes ontology languages and main ontology concepts, especially OWL language, which was used during SLA ontology definition. Section 4 deals with SLA ontology and its internal components. Section 5 discusses about a scenario where semantic QoS negotiation is used. Finally, Section 6 presents conclusions and future work.

2 RELATED WORK

To realize semantic QoS negotiation, as considered in this work, it is necessary to represent QoS information at different perspectives. Using ontology to represent QoS concepts is not a new approach. The FIPA-QoS ontology (FIPA, 2003) deals with QoS at FIPA Message Transport Service. QoS ontology for Workflows and Web Services is described at (Cardoso, 2004). In general, these ontologies are conceived for specific purposes and they usually lack representing of subjective end user QoS.

SLA ontologies are subject of research activities as well. IBM's WSLA (Ludwig, 2003) is a representative initiative to specify SLA parameters associated with services and to identify signatory parties of the contract (provider and customer). However, as traditional approaches, WSLA was mainly defined through technical parameters that are essential for the service configuration and monitoring, but frequently have no meaning for typical users limiting their participation during QoS negotiation. Next section is dedicated to ontology languages, especially OWL (Dean, 2004), which was used to describe SLA ontology proposed.

3 WEB ONTOLOGY LANGUAGES

In computer science, ontology is defined as a software artefact that allows representing and sharing of domain specific knowledge. It can be used by users, databases and applications. The main components of ontology are: concepts, relationships and properties. Concepts represent "things" in a given domain of interest, relationships connect these concepts, and properties characterize concepts attributes.

Existing and new applications can obtain benefits from ontologies. For instance, different documents assigned to the same ontology permit semantic search that considers content instead keywords only. Ontologies can also facilitate communication between intelligent agents through vocabulary sharing. As ontologies are usually expressed in a logic-based language, tools can be used to reasoning about concepts, relationships and properties, inferring new concepts and offering automatic support for intelligent services. Different ontology languages and tools have been proposed. The OWL - Web Ontology Language (Dean, 2004) is a W3C recommendation that was used during SLA ontology development.

3.1 OWL

The OWL language provides three sublanguages designed for use by specific communities of implementers and users. The first one is OWL-Lite that supports classification hierarchy and simple constraint features. The second one is OWL-DL, which supports maximum expressiveness without losing computational completeness. The third one is OWL-Full that supports maximum expressiveness with no computational guarantees.

An OWL ontology consists of Classes, Individuals and Properties. Classes are a concrete representation of concepts in domain we are interested in. Individuals are instances of classes and Properties are binary relations between Individuals. There are two main types of properties: Object properties and Datatype properties. Object properties link an individual to an individual, e.g. *hasSister* property. Datatype properties link an individual to type, e.g. *hasAge* property.

OWL properties have specific characteristics, which derive special properties. For example, each object property may have a corresponding inverse property, e.g. *hasParent* and *hasChild* properties. Other property characteristics include functional, inverse functional, transitive and symmetric. In OWL, properties are used to create restrictions. As the name suggests, restrictions are used to restrict the individuals that belong to a class.

Restrictions in OWL fall into three main categories: Quantifier Restrictions, Cardinality Restrictions and *hasValue* Restrictions. There are two quantifier restrictions: the existential ($\exists$) that can also be read as "*someValuesFrom*" in OWL speak and universal quantifier ($\forall$) or "*allValuesFrom*" in OWL.

4 SLA ONTOLOGY

The QoS negotiation process usually involves technical terms that are essential for providing service but have no meaning for majority of web users. Furthermore, the QoS contract (SLA) commonly reflects this situation. The increase of multimedia application, where users can perceive quality, became clear the inadequacy of this model.

The QoS negotiation model proposal is based in service personalization paradigm (Ribeiro, 2004b), which considers the user QoS desired the start point for service configuration. It also establishes a novel format for QoS contract that explicitly includes subjective quality requirements through which users define QoS. Next sub-sections are dedicated to most important elements of SLA ontology, namely classes and properties. Section 5 depicts the role SLA ontology plays in semantic QoS negotiation.

4.1 Classes

One of the most important elements of an ontology is the concept. In OWL ontology, concepts are described through classes. The concepts in SLA ontology were extracted from ESCHER architecture that is four layer QoS architecture for personalized communication services based on quality of perception (QoP) (Ribeiro, 2004b). According to ESCHER, QoS negotiation depends on two other processes: QoS specification that crosscuts different perspectives (or layers) and QoS mapping that involves these QoS information.

The ESCHER layers are: User, Application, Middleware and QoS Mechanisms. The User Layer is the most abstract perspective in which QoS requirements can be defined. QoS information at this layer represents the QoS level desired by the user, represented by non functional requirements (*NFR*), which are subjective and directly user perceivable, e.g. “performance” and “cost”.

The Application Layer includes technical requirement or *QoS attribute*, used for specialization of subjective user QoS requirements. Applications can be possibly categorized in real-time, interactive, non-interactive and asynchronous classes. Examples of QoS attributes are “frame rate” and “resolution”. At Middleware Layer, the third layer, QoS information is named *QoS characteristic* and it is used for measurement purpose. Examples of QoS characteristics are “delay” and “loss”.

Finally, the most concrete layer of ESCHER architecture is the QoS Mechanism Layer. This is actually an interface for QoS mechanisms that effectively provide QoS guarantees, such as Resource Reservation Protocol (RSVP). At this layer, QoS information is defined as *QoS parameter* used by QoS mechanisms. Figure 1 depicts these QoS concepts, besides other ones that are used to support QoS specification, such as *VirtualResource*, used to abstract physical aspects.

Figure 1: SLA Ontology QoS Concepts.

In ESCHER, the *QoScontract* derived from QoS negotiation and QoS mapping between QoS specifications defined at different layers. The QoS mapping process is helped by information about QoS values (*QoSMapTables*), used during the automatic mapping between user perspective downwards QoS mechanisms, which actually realize QoS guarantees.

The QoS contract is the most important concept defined at SLA ontology. It serves as a basis for the negotiation process and it explicitly describes the QoS requirements stated by all participants of QoS negotiation. As mentioned before, the QoS contract proposal differs from traditional ones by explicit inclusion of user perspective that make it a user-friendly document.

Figure 2 details sub-concepts of QoS contract. There are three components of QoS contract: (1) *UserQoSspec* that includes QoS requirements at user perspective; (2) *ApplicationQoSspec* that represents QoS at application perspective; and (3) *ServiceQoSspec* at service perspective.

The user QoS specification is made up by four elements: (1) QoSdesired that defines high level user requirements, like “High Quality of Audio” and “Medium Cost”; (2) QoSperceived related to the quality perceived by the user according personal characteristics; (3) QoSpriorities that establish priorities in case of impossibility to maintain the QoS agreed, for example, the user can determine that “Cost” is priority rather then “Performance”; and (4) SatisfactionLevel that represents the opinion of user about the quality actually provided and perceived.

The Application QoS specification is dedicated to QoS constraints imposed by the application. This section is composed by three elements: (1) *ApplicationType* that is used to make easier the identification of the type of application by the user, e.g. videoconference and Internet telephony; (2) *Category* that identifies the application by the type of QoS restriction: interactive, non-interactive, real-time and Interactive real-time; and (3) *QoSrequired* that represents a set of constraints imposed by the type of the application.

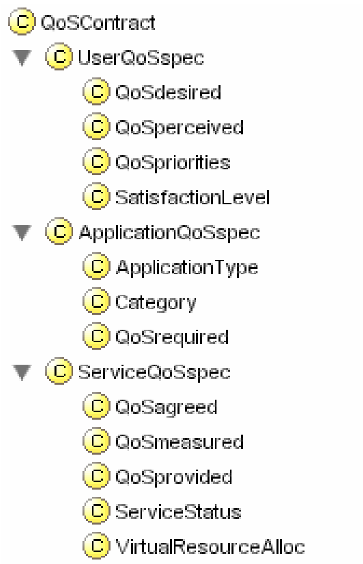

Figure 2: SLA Ontology QoS Contract.

The service QoS specification represents the technical section of SLA, similar to SLS – Service Level Specification (Memenios, 2002). It is composed by five elements: (1) *QoSagreed* that represents the final product of QoS negotiation; (2) *QoSmeasured* a set of pointers that makes possible the evaluation of the QoS agreed; (3) *QoSprovided* that is the actual QoS level provided by network; (4) *ServiceStatus* that indicates the contract situation (Active, Negotiating or Violated); and (5) *VirtualResourceAlloc* represents the amount of resource necessary to realize the service.

The resource negotiation process, as stated by the ESCHER architecture, represents a mapping between user QoS requirements into QoS parameters used by QoS mechanisms, resulting resource allocation. It is important to note that resources are considered virtual resources. This strategy permits to abstract complexity of physical aspects of real resources without losing of expressiveness of QoS specification.

4.2 Properties

There is a consensus about the existence of relationships between QoS under user perspective and QoS under provider perspective. However, the nature of this relation and how it can help the automation of personalized services is not clear. Various properties were defined in SLA ontology to address this aspect. Most of them are related to QoS mapping process, i.e. they state relationships between QoS information defined at different perspectives.

Among properties related to QoS specifications, there are six properties that are directly related to non-functional requirements (NFR), listed at Figure 3. The first two are related to correlation, which is defined in the scope of this work as an implicit relation between distinct NFRs. The nature of correlation can be: conflicting named *hasConflictingCorrelation*, e.g. between "quality of audio" and "usability"; or convergent, named *hasConvergentCorrelation*, such as "performance" and "cost". This characterization concerns the contribution of these requirements to user satisfaction.

The treatment of correlation allows formalizing knowledge about relationship between various aspects that have influence over quality perceived by end users. For instance, experimental results suggest a relationship between performance and cost (Steinmetz, 1995). When they are evaluated together, each of these factors plays a negative influence over the results of another. Figure 3 depicts these NFR related properties, declared as existential constraints.

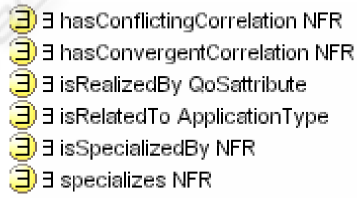

Figure 3: NFR related properties.

The other properties related to NFR are: (1) *isRealizedBy* that relates subjective NFR to more concrete QoS attribute; (2) *isRelatedTo* that relates NFR to the application type; (3) *isSpecializedBy* that relates a more abstract NFR to another, which specializes it; and (4) the inverse property (*specializes*).

There are two properties directly related to QoS attributes. The first one relates QoS attribute to QoS characteristic (*producesImpactOn*). It means that, e.g. changes on QoS attribute "Frame Rate" produce impacts over QoS characteristic "jitter" (delay variation). The second property (*realizes*) is the inverse of *isRealizedBy* property described early. Figure 4 shows these properties.

Only one property that is directly associated to QoS characteristic was defined in SLA ontology. It is the inverse of *producesImpactOn* (see Figure 4), namely *isImpactedBy* property.

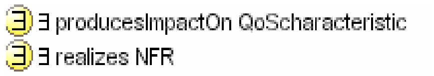

Figure 4: QoS attributes related properties.

Figure 5 presents two properties associated with the type of application (*ApplicationType*) class. The first one assigns application type with the category of application (*Category*), named *isClassifiedAs* property. The second one named *relates* is the inverse property of *isRelatedTo* property assigned to NFR (see Figure 3).

Figure 5: Properties related to Application type.

A partial list of SLA ontology object properties can be viewed through Figure 6. An emphasis is made on *isDirectlyUserPerceivable* property that is associated with user QoS specification. It means that only a subset of NFR that is perceivable by user can be used to specify QoS requirements at user level. The bi-directional line relates property to its inverse.

One of the key features of ontologies that are described using OWL-DL, as in this work, is that they can be processed by a reasoner. A reasoner is a tool that offers some services, such as to check consistency of ontology classes and to automatically compute inferred ontology class hierarchy.

5 SEMANTIC QOS NEGOTIATION SCENARIO

This section deals with an implementation scenario for semantic QoS negotiation. It details what components are involved and how they can be used for service discovery. The architecture proposal was conceived as an extension of Service-oriented architecture (SOA) (Erl, 2005). The original components of SOA remain at extended architecture (see Figure 7): the Service Requestor (or consumer), the Service Provider and Service Repository (or service registry).

Two new components were included at extended SOA architecture proposal: Semantic Web Server and ontologies. Besides the SLA ontology defined at this work, two other ontologies are represented: Web Services Ontologies (WS ontology), such as OWL-S and WSMO, and specific ontology (or ontologies) for NFR.

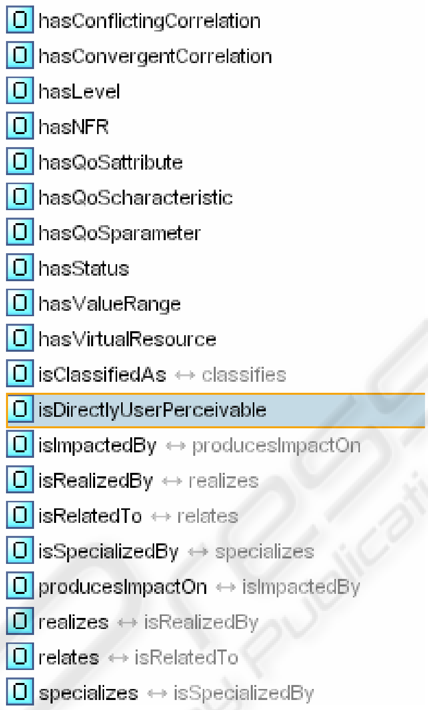

Figure 6: A partial list of SLA ontology object properties.

Semantic Web Services, including Semantic QoS-aware ones, were summarized by the abstract SWS component. At proposed architecture (see Figure 7), mediators are represented by three (possibly composed and complex) mediators (Feier, 2005): Negotiation Mediator concerning QoS Level Mediation; Service Mediator that deals with Web Service Mediation; and Ontology Mediator for QoS ontologies merging or aligning. The concept of mediation considered is generic, representing a software agent that aims to solve conflict situations.

At architecture proposal, SOA activities were modified in order to introduce QoS information. The advertisement (*publish*) of service by the provider (*Service Provider*), in a service registry (*Service Repository*), was improved by including QoS level offered (*QoSoffered*).

The client (*Service Requestor*) tries to discover a service (*find*) at Service Repository, according to user QoS specification (*QoSdesired*) and application QoS requirements (*QoSrequired*). Discovery and selection activities are made by semantic services with mediation capacities and QoS information (*QoSdesired*, *QoSrequired* and *QoSoffered*.

During execution phase (*ServiceStatus Active*), the QoS level agreed can be monitored through network mechanisms (*QoSprovided*), Middleware services (*QoSmeasured*) or human physical senses (*QoSperceived*). If degradation at QoS level is detected (*ServiceStatus Violated*), the service can be suspended and a renegotiation phase is started (*ServiceStatus Negotiating*).

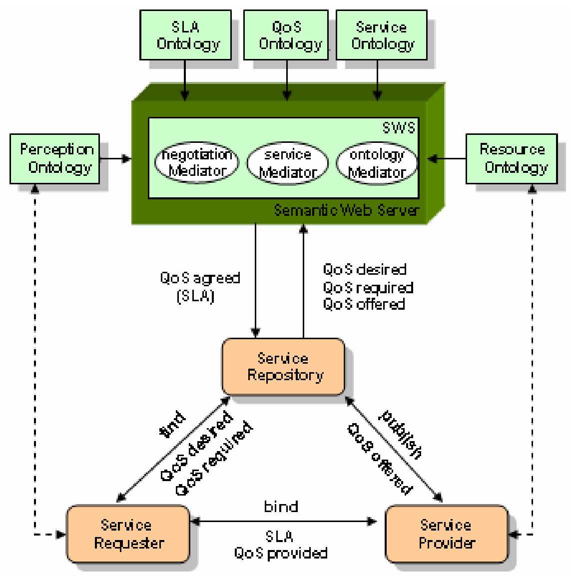

Figure 7: Architecture for Semantic QoS Negotiation.

6 CONCLUSIONS AND FUTURE WORK

Traditionally, QoS negotiation is treated as a low-level network mechanism. However, as the adoption of service-oriented paradigm in application development increases, dealing with QoS negotiation between QoS-aware services has become critical.

The use of ontology in the context of Web Services is relatively a new research topic. OWL-S and WSMO are important references in service ontology field, but the way they describe services are essentially by means of functionality, rather than QoS capabilities. In this paper, an ontological approach for QoS negotiation was detailed. It is based on meaningful QoS contract defined through SLA ontology.

Despite this work had been mainly focused on end users and their participation during QoS negotiation, it can be used by service-to-service approach, i.e. without direct user interference. While the offering of service-oriented application increases, QoS becomes an important aspect to be considered during application development, and then it will be subject of further investigation. Using SLA inside service composition is also subject of future work, as well as the use of mediators to solve potential conflicts and to merge and align ontologies.

REFERENCES

Lawrence Chung, et.al. NonFunctional Requirements in Software Engineering. Ed. Kluwer Academic Publishers (2000).

David Booth. Web Services Description Language (WSDL) Version 2.0 Part 0: Primer. http://www.w3.org/TR/2005/WD-wsdl20-primer-20050803/, (2005)

H. Peter Alesso and Craig F. Smith: Developing Semantic Web Services. A K Peters LTD, (2004).

D. Martin, M. Burstein, O. Lassila, M. Paolucci, T. Payne, and S. McIlraith. Describing Web Services using OWL-S and WSDL. http://www.daml.org/services/owl-s/1.0/owl-s-wsdl.html, (2003)

C. Feier (Ed.): WSMO Primer, WSMO Deliverable D3.1, WSMO Working Draft, latest version available at http://www.wsmo.org/2004/d3/d3.1/, (2005)

Chen Zhou et. al: Web Services Discovery with DAML-QoS Ontology. Int. J. Web Service Res. 2(2): 43-66 (2005)

D.A. Menascé: QoS Issues in Web Services. IEEE Internet Computing, vol. 6, no. 6, (2002)

C. M. F. A Ribeiro, N. S Rosa, P.R.F. Cunha. Meaningful SLA: Towards a QoS Contract based on user perception. Proc. 8th International IEEE EDOCS - Workshop on Contract Architectures and Languages - CoALa (2004)

FIPA. QoS ontology http://www.fipa.org/specs/fipa00094/SC00094A.html, (2003)

J. Cardoso et.al: Modeling Quality of Service for Workflows and Web Service Processes. Web Semantics Journal, Elsevier,1(3): pp. 281-308 (2004)

H. Ludwig, A. Keller, A. Dan, R. P. King, and R. Franck. Web Service Level Agreement (WSLA) Language Specification, v1.0, (2003).

Mike Dean and Guus Schreiber, Editors, W3C Recommendation, OWL – Web Ontology Language Latest version available at http://www.w3.org/TR/owl-ref/, (2004)

C. M. F. A Ribeiro, N. S Rosa, P.R.F. Cunha: Towards a Model for Personalized Communication Services. Proc. of 18th AINA, IEEE Press p. 99-102, (2004)

G. Memenios, G. Pavlou, D. Griffin, and L. Georgiadis. Service Level Specification Semantics and Parameters. Internet Draft, tequila-sls-02, (2002)

Ralf Steinmetz and Klara Nahrstedt: Multimedia: computing, communications, and applications. Prentice Hall, (1995).

Thomas Erl: Service-Oriented Architecture : Concepts, Technology, and Design, Prentice Hall (2005)