

A semantic approach for QoS specification of communication services using QoE parameters

José Cé Júnior · Achilles C. Prudêncio ·
Roberto Willrich · Madalena P. da Silva

Received: 20 April 2011 / Accepted: 30 October 2012 / Published online: 21 November 2012
© The Brazilian Computer Society 2012

Abstract Various operations related to Quality of Service (QoS) management in communication networks require the clients/users to specify the quality level of the network service. The current QoS proposals adopt a fixed set of performance parameters at the network level to specify quality levels, avoiding dealing with the problem of heterogeneity among QoS parameters and metrics during the service management. However, in several situations where humans are the end-users of the service, the quality level should be specified using Quality of Experience (QoE) parameters, more natural for humans than network performance parameters. On the other hand, the QoS specification using QoE parameters is not sufficient to the service providers; they must translate QoE parameters into network parameters. This paper proposes a semantic approach to the automatic QoE/QoS mapping using the Network QoS Ontology (NetQoSOnt), offering automatic and extensible translation between QoE and QoS parameters. The use of our proposal is illustrated by supporting a voice over IP service negotiation.

Keywords Quality of Service · Ontology · OWL · SWRL

A previous version of this paper has appeared at Webmedia 2010, the Brazilian Symposium on Multimedia Systems and the Web.

J. Cé Júnior (✉) · A. C. Prudêncio · R. Willrich · M. P. da Silva
Department of Computer Science and Statistics (INE),
Federal University of Santa Catarina (UFSC), Florianópolis, Brazil
e-mail: jcjr@inf.ufsc.br

A. C. Prudêncio
e-mail: achilles@inf.ufsc.br

R. Willrich
e-mail: willrich@inf.ufsc.br

M. P. da Silva
e-mail: madalena@inf.ufsc.br

1 Introduction

New networked applications and customers consider essential that service providers guarantee the quality of the services offered. Otherwise, customers may be dissatisfied with the service. This has stimulated significant research and implementation efforts in Quality of Service (QoS) at the application, operating system and network levels. In a QoS-aware system, a network service provider (NSP) negotiates a contract with clients. These contracts, known as service level agreements (SLAs), specify several service parameters, including the QoS level to be guaranteed. An SLA contains a list of service level specifications (SLSs) that specify the quality level of a service using a set of technical parameters and their values.

Nowadays, there is neither a standardized SLA/SLS specification template, nor a universally adopted set of QoS parameters (and their metrics). In general, the current QoS solutions adopt performance metrics at the network level, such as OWD (One Way Delay), IPDV (IP Packet Delay Variation) and PLR (Packet Loss Rate). Thus, these network QoS parameters are used to define the network's capability to meet the requirements of customers and applications.

Some works have shown that adopting parameters determining perceived service quality and the user satisfaction level is more natural for humans than network QoS parameters [1,2]. The perceived service quality as well as the user satisfaction level can be qualified in terms of QoE (Quality of Experience) parameters. MOS (Mean Opinion Score) [3] is an example of a QoE metric that has been frequently used to assess the perceptual QoS in voice over multimedia applications. For instance, it is more “natural” for people to express quality of a VoIP service using MOS than network QoS parameters.

The use of QoE parameters to specify the service quality poses some challenges related to the QoS mapping: the QoE parameters must be understood and translated into quantitative QoS parameters that can be recognized and controlled by the NSPs [4,5]. This is necessary because the NSP must know the network QoS parameters to configure its network (and link data layer devices). Two of the challenging issues of the QoS/QoE mapping are: (i) there is an expanding list of QoE parameters, and (ii) different NSPs can adopt different network performance parameters to specify QoS requirements. These issues can be summarized by heterogeneity and extensibility issues.

In this paper, we intend to make a contribution so that NSPs, standardization organizations and application developers can publish new QoE/QoS parameters and related mapping rules and these new parameters can be automatically used during the various operations related with the QoS management in network services. Here, there are two important requirements that such a solution must meet: (i) it must offer mechanisms allowing the sharing of new QoE/QoS parameters, metrics and mapping rules based on standard languages; and (ii) it must offer mechanisms that allow an NSP to import these new parameters, metrics, and rules, and autonomously consider them during the QoS management. We consider that the Semantic Web technologies, such as Ontologies and Reasoning, are natural solutions that satisfies these requirements.

In the Web service domain, heterogeneity and extensibility issues must also be handled during the service selection and negotiation [6]. In the Semantic Web Services, these issues are dealt with by means of ontologies, which can be defined as sets of classes (concrete representation of concepts), properties (or relations), individuals (or instances) and axioms that can be used to describe a domain [7]. The Web Ontology Language (OWL) [8] is the W3C language standard designed for coding ontologies. Ontologies provide a formal mechanism that can bridge the gap between different terminology and semantic related metrics. In particular, ontologies have as characteristics intelligibility, interoperability, transparency and extensibility. A Knowledge Base (KB) is a special database for knowledge management providing means for information to be collected, organized, shared, searched and utilized. Semantic reasoners are able to infer logical consequences given a set of axioms or facts in the KB.

Some works have been using ontologies in the network services domain. In [9], the authors point out that ontologies can be used as a formal specification of network services management semantics required as the building blocks to create reasoning mechanisms to allow developing self-managed network service providers. In [10], Zrelli et al. adopt ontologies as an alternative to deal with the challenges that organizations face in managing large scale multi-technology and multi-vendor heterogeneous network infrastructures. Ontologies are used as mechanisms to unify concepts to

provide interoperability between systems and organizations. The ontologies are also being applied in the Internet of Things (IoT) [11] to address the heterogeneity of IoT components.

In [12], we proposed an OWL-based QoS Ontology for network services, called NetQoSOnt. Using this ontology, it is possible not only to define a common vocabulary but also to extend this vocabulary with new QoS parameters and metrics. This ontology allows the expression of equivalence relations between QoS specifications in different levels, including the expression of equivalence relation between QoS specifications using the network performance parameters (Internet layer) and specifications using QoE parameters (user layer). For instance, NetQoSOnt allows the user to specify that the quality level $MOS \leq 4$ is equivalent to guaranteeing certain upper bounds for OWD, IPDV and PLR at the Internet layer. However, as it will be discussed later, this form of QoS/QoE mapping has a granularity issue.

This paper proposes a semantic approach, which reuse the NetQoSOnt ontology, providing the functionalities necessary to the automatic mapping of QoS/QoE specifications into different network levels. The solution proposed in this paper is to specify the QoS/QoE mapping using rules written in the SWRL language [13]. Consequently, our semantic approach offers flexibility in terms of QoS/QoE parameters and an improvement on the granularity issue. Our semantic approach allows that different service providers, clients and organizations in general can express their quality levels (required or guarantee qualities) in terms of arbitrary parameters and metrics according to their needs.

Thanks to the automatic classification provided by formal ontologies, our semantic approach makes it possible to compare these quality specifications despite the heterogeneity among QoE/QoS parameters. Moreover, our semantic approach offers flexible extensibility, requiring only that service providers and standardization organization publish new parameters, metrics and respective mapping rules specified as classes and individuals in an OWL file on their Web sites.

The rest of this paper is organized as follows: Sect. 2 reviews concepts related to QoS and QoE. Section 3 describes and analyzes related work and shows the advantages of the proposed approach. Section 4 presents the NetQoSOnt ontology. Section 5 presents the proposed extension of NetQoSOnt allowing the specification of QoE/QoS mapping using SWRL rules. In Sect. 6, the use of our semantic approach to automatically map QoS/QoE parameters is illustrated and tested in the VoIP domain. Finally, Sect. 7 presents conclusions and future work.

2 QoS and QoE

ITU-T Recommendation E.800 [14] defines QoS as “the collective effect of service performances, which determine the

degree of satisfaction of a user of the service”. In network communication services, instead of specifying the degree of satisfaction experienced by the users (i.e., QoE parameters), the QoS specification normally is described by network performance parameters. This section presents some basic concepts in QoS and QoE, including the QoS/QoE mapping.

2.1 QoS in network services

As previously described, in a QoS-aware network system, an NSP negotiates SLAs with its clients, each SLA containing a list of SLSs specifying the quality level of a specific service or traffic. In general, each SLS defines the quality level negotiated for a specific network traffic via a set of network performance parameter/value (or value range) pairs.

Because of the lack of standardization in terms of QoS parameters, NSPs can adopt SLSs using different QoS parameters and metrics. The choice of these parameters can be driven by various factors such as network technology, QoS solution and marketing strategy. An attempt to compile a standard set of SLS parameters was made by the TEQUILA project [15]. Like most works, the TEQUILA project adopted network performance parameters to specify QoS, including OWD, PLR and IPVD.

For instance, consider a hypothetical NSP adopting the Differentiated Service (DiffServ) architecture and a set of Class of Services (CoS), each one satisfying specific network performance requirements specified for the Y.1541 QoS classes [16]. Thus, each QoS class must guarantee determined maximum values of IPTD, IPTV and PLR. So, it is natural that this NSP adopts these network parameters during the definition of its SLS template. During the service negotiation, the NSP could associate each SLS with the CoS that meets the required quality. Other NSP adopting a different QoS solution can adopt different QoS parameters (e.g., cell loss ratio, maximum cell transfer delay) and different measurement methods.

2.2 QoE

There are several definitions of QoE in the literature. ITU-T [17] defines QoE as “the overall acceptability of an application or service, as perceived subjectively by the end-user”. It is influenced by all the components involved in end-to-end communication, including the user, the terminal, the communication network and the service infrastructure.

Nowadays, there are various QoE parameters and metrics in the literature that can be used in different applications and media types. MOS (Mean Opinion Score) [3] has been widely used to assess the perceptual QoS in audio/video applications, such as VoIP, video conference [18,19] and TV over IP [20]. MOS gives a numerical indication of the

perceived quality of the media, expressed on a scale from 1 to 5: 1—bad, 2—poor, 3—fair, 4—good, 5—excellent.

Because MOS testing is subjective and requires carefully prepared and controlled test conditions that are difficult and expensive to set up and execute, some objective methods were proposed. In the VoIP domain, E-Model [21] and PESQ [22] are examples of objective measurement of quality. The output of the E-Model algorithm is a quality rating value varying from 0 to 100, called R factor, which is defined in [21] as

$$R = R_o - I_s - I_d - I_{e\text{-eff}} + A, \quad (1)$$

where R_o is the basic signal-to-noise ratio (including noise sources), I_s is impairments simultaneous to voice signal, I_d is impairments caused by delay and the effective equipment impairment, $I_{e\text{-eff}}$ represents impairments caused by low bit-rate codecs and due to randomly distributed packet losses, and the advantage factor A allows for the compensation of impairment factors when there are other advantages of access to the user. Among all these factors, only I_d and $I_{e\text{-eff}}$ are affected by network conditions. Taking the default values for the other factors (as defined in [23]), the factor R can be defined by

$$R = 93.2 - I_d - I_{e\text{-eff}}. \quad (2)$$

Cole and Rosenbluth [24] propose a simpler linear function to estimate the values of I_d , given by

$$I_d = 0.024d + 0.11(d - 177.3)H(d - 177, 3), \quad (3)$$

where d is the end-to-end delay (in milliseconds) and $H(x)$ is the Heavyside function:

$$H(x) = 0 \text{ if } x < 0; H(x) = 1 \text{ if } x \geq 0.$$

$I_{e\text{-eff}}$ can be calculated using the Eq. [21]

$$I_{e\text{-eff}} = I_e + (95 - I_e) \frac{Ppl}{\frac{Ppl}{BurstR} + Bpl}, \quad (4)$$

where Ppl is the packet-loss probability, Bpl is the packet loss robustness factor, and $BurstR$ is the so-called burst ratio. When packet loss is random (i.e., independent), $BurstR = 1$, and when packet loss is bursty (i.e., dependent) $BurstR > 1$.

The R value can be used to estimate MOS. Equation 5, presented in [21], defines how to map an R value into a MOS rate. Table 1 presents the relationship of R-factor values to MOS and to the subjective quality rating. I_e and Bpl are codec-dependent factors whose values for some codecs are defined in [23].

$$\text{MOS} = \begin{cases} 1, & \text{if } R < 0 \\ 1 + 0.035R + R(R - 60) \cdot (100 - R)^7 \cdot 10^{-6}, & \text{if } 0 < R < 100 \\ 4.5, & \text{if } R > 100 \end{cases} \quad (5)$$

Table 1 Relation between *R*-value, MOS and subjective quality (adapted from [21])

<i>R</i> -value	MOS	Subjective quality
≥90	≥4.34	Best
≥80	≥4.03	High
≥70	≥3.60	Medium
≥60	≥3.10	Low
≥50	≥2.58	Poor

From the user's point of view, it is more convenient to describe the quality level using QoE parameters than network performance parameters. For example, during a VoIP service negotiation, it is more natural for customers to express their quality needs in terms of MOS score than network parameters, like OWD, PLR and PVD. However, there exists a large and expanding list of QoE parameters and metrics. An ideal QoS solution should offer means to specify the quality level with flexibility in terms of QoS parameters.

2.3 QoE/QoS mapping

As already presented, the perceived quality is influenced by all the components involved in the end-to-end communication. The network service is sometimes the only component that is controlled by the NSP. On its side, the NSP must set-up the network devices in order to guarantee a network quality level sufficient to ensure the QoE required by the customer. This quality level can be calculated considering the default values for the other components impacting the QoE (e.g., human factors, equipment quality).

In order to allow the quality level of a network service to be expressed using QoE parameters, one of the first actions to be performed by the NSP is the QoS/QoE mapping. Two of the challenging issues of the QoS/QoE mapping are: (i) there are several QoS/QoE parameters and metrics used in different media and services, and the list of QoE parameters is in continuous expansion; and (ii) different NSPs can adopt different network QoS parameters to specify QoS (according to its network technologies, QoS solutions and marketing strategy).

There are some works that aim at mapping QoE parameters and network performance parameters. In [4,5,25], the authors propose mathematical expressions for QoS/QoE mapping. Ghinea and Thomas [5] identified a relation of proportionality between QoE and QoS, where a QoE score is determined by a weighted sum of bit error, segment loss, order, delay and jitter metrics. The weight associated with each metrics was empirically derived based on subjective tests. Siller and Woods [4] propose that a normalized QoE can be defined as some function of network and application

QoS metrics. In this work, the Network QoS metric is defined as the sum of the weighted factors given by three network metrics: delay, jitter and packet loss.

3 Related work

The QoE/QoS mapping solutions presented in the last section are not sufficient to deal with the heterogeneity and extensibility issues present in QoS management. As previously presented, two significant and challenging issues are yet to be fully addressed: (i) there is an expanding list of QoE parameters, and (ii) different NSPs can adopt different network performance parameters to specify QoS requirements. An ideal QoS management solution should support both the QoS specification allowing the use of an extensible list of QoS parameters and the automatic mapping of these parameters to those understood and controlled by the NSPs.

Semantic approaches to QoS specification using a flexible list of parameters are not a new matter in Web Services (WS). Four important contributions in the QoS for WS are OWL-QoS [26], QoSOnt [27], Service Level Ontology [28] and OWL-Q [29]. These contributions use OWL version 1.0 and, as discussed further in [12], this version of OWL does not provide sufficient support for data type constraints, which are necessary to correctly model parameter bounds. The QoS ontologies cited try to bypass this limitation using OWL object properties, custom XML and semantic Web rule languages. For instance, OWL-Q [29], an extensible ontological specification for semantic QoS-based WS description, bypass this limitation by extending OWL with SWRL rules reasoning about relations between properties in order to enforce constraints. Note that the use of SWRL rules to specify the QoE/QoS mapping is not addressed in any of the works above.

There are some works that consider ontologies and QoE/QoS mapping in the network domain. For instance, Gallo et al. [30] propose a QoE ontology used by an agent-based platform to map quality of service to experience in conventional and active networks. Differently from our approach, the QoE ontology proposed by [4] specifies the vocabulary and communication messages of the agents composing the proposed platform to map quality of service to experience. Therefore, the QoE/QoS mapping itself does not follow a semantic approach.

In [10,31,32] the authors propose semantic approaches to deal with network monitoring, authorization of explicit QoS services invocations and to manage large-scale networks, respectively. In these three works, the ontologies specify QoS using a fixed list of network parameters. Therefore, these approaches also do not deal with the heterogeneity and extensibility issues. Moreover, these ontologies do not deal with QoE/QoS mapping.

In [33], Macián et al. propose a framework based on ontologies and rules to perform the QoE/QoS mapping automatically. This framework combines SWRL rules and a language that represents the semantics of mathematical objects (such as OpenMath [34]) to obtain the semantic representation of the relationship between the technical component of QoS and QoE. A semantic reasoner will use these rules and the instances of the ontology to generate new instances related to the QoE domain providing information about the QoE level. However, the proposed framework was not tested or applied in [33]. Moreover, this work does not deal with the heterogeneity and extensibility issues.

In [35] describes an ontology-based SLA formalization using OWL and SWRL in the telecommunication domain. Six generic ontologies were designed to provide a framework for describe SLAs: Unit Ontology, Time Unit Ontology, Temporal Ontology, Concurrency Ontology, Network Metrics Ontology and SLA ontology. However, the layer abstraction necessary to deal with mapping between QoS parameters in different level is unavailable.

In [36], Chen et al. propose a QoS model for software services deployed in the cloud. In this work, an ontology is used to describe QoS concerns from three dimensions: system resource utilization, service performance, and price. An ontology language is used to describe these concerns and their properties. However, like the previous one, this ontology lacks an abstraction to model the system layers. Moreover, the satisfaction of the system is defined as a weighted sum of parameter satisfaction, a non ontology-based method.

In [9], Rodrigues et al. the authors define an ontology for multiservice IP networks targeting multiple service management goals. The authors are concerned with the QoE/QoS mapping. This mapping is driven by a so-called Multiservice Monitoring. However, the QoE/QoS mapping is not detailed in [9]. Moreover, only network performance parameters are specified in this ontology.

Fallon and Sullivan [37] work in building, populating and evaluating an ontological Knowledge Base (KB) for quality management in IPTV services. This KB represent end-user service experience and context. The terminal reporting is used to collect the QoS and QoE metrics that are semantically mapped and stored in the KB. That knowledge is subsequently used to analyze and optimize the quality of IPTV service delivery. The QoE/QoS mapping is not considered in this work; all quality metrics are classified as individuals (Metrics). Therefore, these two concepts are not distinguished in the KB.

Alípio et al. [38] propose an ontological representation of network services to create a common vocabulary, including a service classification, and to map service requirements into network configuration. However, [38] does not explain

how this mapping is performed. Moreover, this ontology is specific for Diffserv and deals only with network level parameters.

The new version 2.0 of OWL provides syntactic support to define numeric ranges, backed up by reasoner support (in reasoners like FaCT++ and Pellet). Using OWL 2.0 makes the previous QoS/QoE mapping solutions obsolete. In [12], we present NetQoSOnt, an ontology that allows the semantic interoperability in terms of parameters used in the QoS specification of network services. This ontology is being developed utilizing OWL 2.0 [8] and uses the characteristics of this new version of OWL for the comparison of QoS parameters through inference.

In this paper we extend NetQoSOnt with SWRL rules in order to specify QoE/QoS rules. Note that we do not intend to propose new methods to map QoS and QoE, as proposed by [4, 5, 25]. The novel aspect of our approach is that it makes it possible that clients and NSPs express their quality level using different QoE and QoS parameters. The existing and future QoE and QoS parameters and mapping methods can be specified as new OWL classes and SWRL rules. After published as an OWL file, these concepts can be imported into the Knowledge Base (KB) of the NSPs. Thus, this new knowledge can be applied to automatically map QoE and QoS parameters and, using the new features of OWL 2.0, the QoS specification using different QoE/QoS parameters can be compared.

4 NetQoSOnt

This section presents NetQoSOnt [12], a QoS ontology oriented to network services able to provide a formal, extendable and machine-processable specification to state QoS specifications. NetQoSOnt is designed to be a basis for semantic-based approaches for all stages of QoS service management, including selection, negotiation, dynamic invocation, and monitoring of network services. New ontologies importing concepts of NetQoS-Ont are necessary for any QoS ontology-based operation.

4.1 NetQoSOnt modules

As shown in Fig. 1, NetQoSOnt classes are organized in modules. Each module involves classes of one of the four layers of the TCP/IP stack. Additionally, there is a module designed to contain QoS specifications at the user's level and another one containing basic concepts that are reused by all other modules. The notion of layers is very important to model QoS in network services because the quality parameters of a layer may (and usually do) depend on the quality parameters of the lower layers.

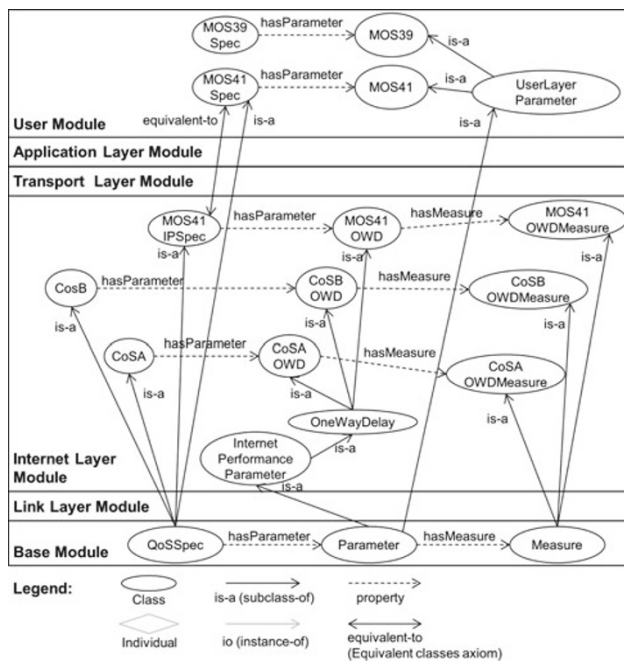


Fig. 1 The NetQoSOnt modules

4.1.1 Base module

The Base module was defined to aggregate a list of generic resources necessary to create QoS specifications and parameters. This module is composed of various classes, the most important of which are illustrated in Fig. 1. The class *Parameter* represents a measurable property in a network layer. *QoS Spec* is the super concept of any QoS specification, i.e., a QoS specification can be specified as a subclass (specialization) of *QoS Spec*. A QoS specification is defined as a set of QoS parameters, identified by the property *hasParameter*.

NetQoSOnt reuses the unit concepts from the Measurement Units Ontology (MUO) [39]. The class *Measure* is a subclass of *QualityValue*, a concept from MUO representing the value of an individual quality, for instance, 150 ms for OWD. Each *Parameter* is related to a *Measure*, specified by the property *hasMeasure*.

4.1.2 Link layer module

This module aggregates concepts related to the link layer of the TCP/IP stack. For instance, the concept *LinkPerformanceParameter* has been derived from *Parameter* to define parameters specific to this layer. For example, Fig. 2 presents two performance parameters of the 802.16e standard, Tolerated Jitter and Maximum Latency, specified as subclasses of *LinkPerformanceParameter*, and they refer to a *Measure* subclass that defines the unit (bits per second and per millisecond, respectively), and their respective values.

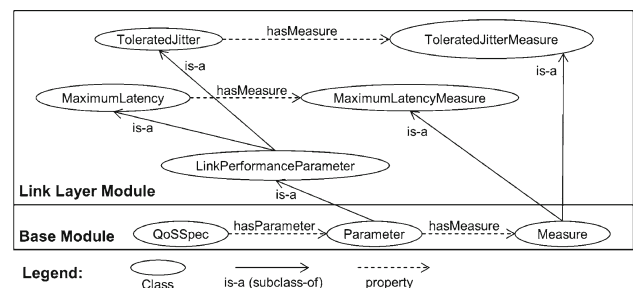


Fig. 2 The Link Layer module

4.1.3 Internet layer module

The Internet layer module specifies concepts related to the Internet layer of the TCP/IP stack. The QoS parameters at the Internet layer are derived from *InternetPerformanceParameter*. The parameters PLR, OWL and PDV have been defined based on the IETF RFCs 2680, 2679 and 3393, respectively. Figure 1 presents only *OneWayDelay* for the sake of simplification.

Adopting NetQoSOnt, the NSP must maintain the quality level offered by their network services in its KB. To illustrate the use of NetQoSOnt, suppose that an NSP adopts a QoS solution based on two classes of service: CoSA that satisfies the Y.1543 QoS Class 0 criteria [16] ($OWD \leq 100$ ms, $IPDV \leq 50$ ms, $PLR \leq 0.001$) and CoSB that satisfies the Y.1543 QoS Class 1 criteria ($OWD \leq 400$ ms, $IPDV \leq 100$ ms, $PLR \leq 0.001$). The team of network specialists of this NSP must produce a subclass of *QoS Spec* for each CoS using an ontology development tool (e.g., Protégé [40]). Figure 1 illustrates the specification of the QoS level offered by CoSA and CoSB that must be maintained in the KB, represented by the classes *CoSA* and *CoSB*. Again, this figure presents only the parameter OWD, where the OWD limit of CoSA is defined by *CoSAOWDMeasure* as ≤ 100 ms, and the OWD limit of CoSB is defined by *CoSBOWDMeasure* as ≤ 400 ms.

4.1.4 Transport layer module

This module specifies concepts related to the transport layer of the TCP/IP stack. For instance, the QoSTP protocol [41] allows the configuration of the congestion window algorithm and the error control algorithm. These parameters can be modeled by means of a specialization of the *Parameter* concept.

4.1.5 Application layer module and user module

Similarly to the previous layers, NetQoSOnt provides the means to specify QoS specifications, parameters and met-

rics at the application and user layers. Based on the concepts defined in our ontology, application developers and organizations can publish in their websites new concepts, for instance, to express QoE parameters. For example, using NetQoSOnt a hypothetical VoIP standardization organization can publish a set of ranges of MOS scores. In Fig. 1 only one range is represented, $MOS \geq 4.1$, specified by *MOS41Spec* (a subclass of *QoSSpec*) at the User Layer.

Moreover, customers and users can express their quality level needs as subclasses of *QoSSpec*. For instance, during a service invocation a user can request a VoIP call with quality $MOS \geq 3.9$. This request is specified by *QoSSpec MOS39Spec*, presented in Fig. 1. In [42], we propose a SIP (Session Initiation Protocol) extension for specifying QoS requirements using NetQoSOnt. Because there is no subclass of *QoSSpec* specifying $MOS \geq 3.9$ in the KB of the NSP, the negotiation system must include the class *MOS3.9* in the KB.

4.2 Equivalence between QoS specifications and the inference process

Figure 1 shows also an example of how NetQoSOnt specifies equivalence relations between QoS specifications. Returning to our hypothetical VoIP standardization organization, in addition to the subclasses of *QoSSpec* specifying a set of ranges of MOS scores, this organization must publish the quality level that must be guaranteed at the Internet layer to ensure each range. The *MOS41IPSpec* in the Internet layer specifies the QoS level that the network service should ensure to allow a $MOS \geq 4.1$. For simplification, Fig. 1 presents only one of the QoS parameters used to specify this network QoS level, the OWD parameter (although not showed here, other ranges of MOS score and QoS parameters were defined). As seen in Fig. 1, *MOS41IPSpec* has *MOS41OWD* as one of these properties, which specifies the OWD limits to guarantee $MOS \geq 4.1$. The measurement of OWD is defined in *MOS41OWDMeasure* as ≤ 150 ms. To define the equivalence between *MOS41Spec* at the user level and *MOS41IPSpec* at the Internet level, an equivalence relation of classes in ontologies is used (*equivalent-to*).

Suppose again that a customer of an NSP requests a VoIP service with quality $MOS \geq 3.9$ and the NSP adopts a QoS solution based on two CoS: CoSA and CoSB. To accept or reject the customer's request, the NSP must first know if there is a CoS that meets the customer's request. To answer this question the NSP uses a semantic reasoner that tries to derive answers from the KB. The reasoner can recognize which concepts fit under which definitions and, therefore, maintain the hierarchy correctly. The semantic reasoner can maintain the hierarchy of the subclasses of *QoSSpec* when they are applied to the previous example of service negotiation.

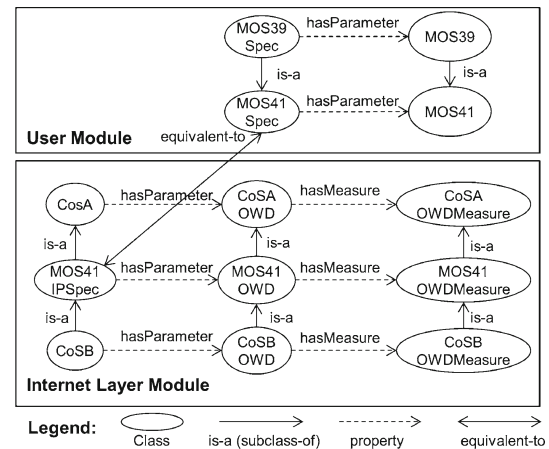


Fig. 3 The inferred class hierarchy

Figure 3 presents the inferred class hierarchy. Since $3.9 \leq 4.1$, and the units are the same, the reasoner will infer that *MOS41* is a subclass of *MOS39*, and consequently *MOS39Spec* will be inferred as a subclass of *MOS41Spec*. Since $100 \leq 150 \leq 400$, *CoSAOWD-Measure* will be inferred as a subclass of *MOS41OWD-Measure*, which in turn is a subclass of *CoSBOWD-Measure*. Consequently, *CoSAOWD* will be inferred as a subclass of *MOS41OWD*, which will be a subclass of *CoSBOWD*. Moreover, *CoSA* will be inferred as a subclass of *MOS41IPSpec*, which will be a subclass of *CoSB*. Finally, as the OWD limit of CoSA is less than that specified by *MOS41IPSpec*, *CoSA* is inferred to be more specialized than *MOS41IPSpec*.

In conclusion, *CoSA* is a subclass of *MOS39Spec* and therefore CoSA meets the customer requirements. Note that to accept the user request, the NSP must also use others parameters (e.g., source and destination addresses, traffic conformance) and verify the current resource availability.

4.3 QoE/QoS mapping

NetQoSOnt does not allow the specification of QoS mapping rules, only the definition of equivalence relations between QoS specifications. This limitation creates problems. First, the equivalence relations in Net-QoSOnt are biconditional logical connective between statements, i.e., a QoS level in a specific layer (described by a subclass of *QoSSpec*) is met if and only if an equivalent QoS specification is guaranteed in a lower layer. For instance, *MOS41Spec* is equivalent to *MOS41IPSpec*, where the latter specifies the quality level as $OWD \geq 150$ and $PLR \geq 0.03$. However, according to Eq. 2 (Sect. 2.3), there are other combinations of delay and packet losses that ensure the same MOS score.

The second problem is related to network resource optimization. Taking the previous example into consideration, when a hypothetical standardization organization publishes

the MOS parameter using NetQoSOnt, it must also publish a set of equivalence relations between MOS ranges and QoS specification at the Internet level. Consider that two MOS ranges are published: $MOS \geq 4.1$ is equivalent to guaranteeing $OWD \leq 150$ ms, $IPDV \leq 50$ ms, $PLR \leq 0.03$; and $MOS \geq 3.6$ is equivalent to guaranteeing $OWD \leq 350$ ms, $IPDV \leq 125$ ms, $PLR \leq 0.15$. Using the semantic reasoner, the NSP can identify which QoS level guaranteed by CoSA ($OWD \leq 100$ ms, $IPDV \leq 50$ ms, $PLR \leq 0.001$) is equivalent to $MOS \geq 4.1$. Note that the effective MOS range guaranteed by a CoS is not calculated.

In this case, if a certain CoS can ensure $MOS \geq 3.9$ (based on its maximum limits of OWD, IPDV and PLR), using only the equivalence relations the reasoner will infer that the MOS range guaranteed by this class is $MOS \geq 3.6$. In this case, if a customer request $MOS \geq 3.9$ for its VoIP traffic, the NSP will classify this traffic in CoSA. As previously described, this conclusion was reached from the observation that QoS specification $MOS \geq 3.9$ was classified by the reasoner as a subclass of QoS specification $MOS \geq 4.1$ that in turn is equivalent to the QoS specification guaranteed by the class of service CoSA. The rationale is that if the NSP guarantees resources to ensure quality $MOS \geq 4.1$, it also ensures that the user's request of $MOS \geq 3.9$ is honoured. In any case, if there was a way to dynamically generate the MOS range guaranteed by a CoS, then it could be concluded that a less expensive class of service could satisfy the customer needs.

The problems pointed out above arise due to a lack of granularity of the equivalence relations between QoS specifications. In this paper, we address this issue of lack of granularity by offering functionalities for supporting the automatic mapping between QoS parameters adopted by the NSP and the QoS/QoE parameters adopted by the customer/user. In NetQoSOnt, the equivalence relations between QoS specifications are statically defined. In this paper, these equivalence relations can be created automatically when the customer/user uses a new QoS/QoE parameter. For instance, when the customer requests $MOS \geq 3.9$, and MOS is an unknown parameter of the NSP, they can import this parameter and its mapping rules, and automatically calculate the quality level offered by its network services using now the MOS score.

5 Extending NetQoSOnt with SWRL rules

This section presents an extension of the NetQoSOnt ontology seeking to overcome the previously cited limitations during the QoS/QoE mapping. This extension defines new concepts and individuals necessary to specify formally the QoS/QoE mapping, including a new module called QoS/QoE Mapping. Particularly important is that the proposed extension

allows the automatic QoS/QoE mapping using SWRL rules stored in individuals of the ontology.

5.1 New concepts and individuals

In NetQoSOnt [12], QoS specifications are described by specializations of the class *QoSSpec*. In the present proposal, QoS specifications that are guaranteed by the network services offered by the NSPs (e.g., QoS specifications describing the quality level guaranteed by each CoS) are described as individuals, while customer/user's QoS requirements are described by specializations of the class *QoSSpec*. This interpretation is necessary because it is not possible to associate SWRL rules with classes, only with individuals.

Figure 4 illustrates the use of the proposed NetQoSOnt extension in the VoIP domain. For purposes of simplification, this figure does not present all the NetQoSOnt modules. Two concepts in the application layer module are presented in this figure:

- *ApplicationParameter* is a subclass of *Parameter* to define parameters at the application layer;
- *Codec* is a subclass of *ApplicationParameter* used to specify the codec chosen by the customer/user of a VoIP service. An individual belonging to the class *Codec* represents a specific codec that is identified by its RTP Payload Type (PT) value [43]. PT values are specified by subclasses of *Measure*, using the object property *hasMeasure* and the data property *qualityLiteralvalue*.

The user layer module aggregates concepts related to QoE. Figure 4 presents three examples of subclasses of

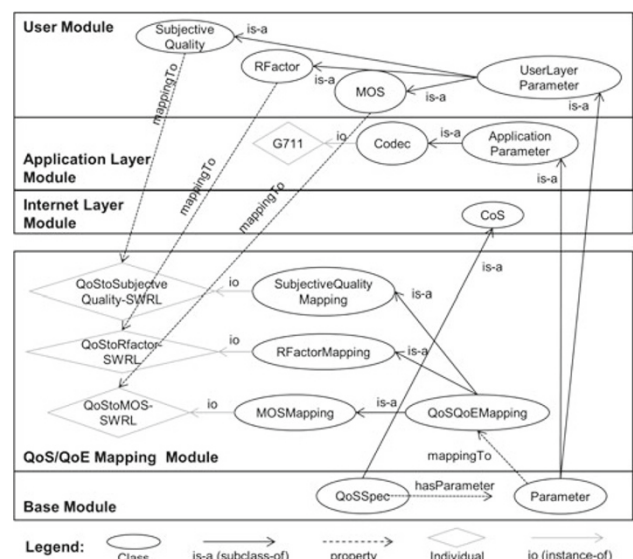


Fig. 4 Specifying QoE concepts in the VoIP domain

UserLayerParameter in the VoIP domain: *MOS*, *RFactor* and *SubjectiveQuality*. These classes allow the users to specify its required QoE through MOS, R factors and subjective quality values. The subjective quality levels and the relationships between these parameters are presented in Table 1.

In the Internet layer module, the class *CoS* (a subclass of *QoSSpec*) has been defined to allow the NSPs to specify the quality level guaranteed by their CoSs. The NSP should include in its KB an individual belonging to the class *CoS* for each CoS adopted by the NSP. As will be presented in this paper, to each individual *CoS* can be associated, with the object property *hasEquivalent*, a set of individuals *QoSSpec* specifying qualities levels equivalent to this CoS. These individuals *QoSSpec* will be automatically generated during the QoS/QoE mapping.

5.2 QoS/QoE mapping

As presented in Sect. 4.3, in NetQoSOnt the equivalence relations between QoS specifications are statistically defined. For instance, NetQoSOnt makes it possible to guarantee that a certain QoS level at the network layer is equivalent to guaranteeing a certain QoE level. However, NetQoSOnt does not allow the specification of how a network QoS specification should be mapped to certain QoE level. In order to offer flexibility in terms of QoS/QoE mapping, in this paper we extended NetQoSOnt with a new module, called QoS/QoE Mapping.

In the QoS/QoE Mapping module, new subclasses of *QoSQoEMapping* can be defined to describe new QoS/QoE mapping rules. Figure 4 presents three subclasses of *QoSQoEMapping* defining three QoS/QoE mapping rules in the VoIP domain: *RFactorMapping*, *MOSMapping* and *SubjectiveQualityMapping*. In addition, one individual for each one of these subclasses is presented: *QoSToMOS-SWRL*, *QoSToRFactor-SWRL* and *QoSToSubjectiveQuality-SWRL*. Each one of these individuals keeps QoS/QoE mapping rules using the SWRL language. As illustrated in Fig. 4, the relationship *mappingTo* links the class *Parameter* to individuals belonging to the class *QoSQoEMapping* that specify mapping rules to each parameter.

The mapping rules present in the individual's *QoSToRFactor-SWRL* and *QoSToSubjectiveQuality-SWRL* were described based on Eqs. 2, 3, 4 and Table 1. Figure 5 presents one of the mapping rules maintained in the individual *QoSToMOS-SWRL*. These rules allow the estimation of a MOS score based on the OWD and PLR parameters (i.e., these rules are an SWRL representation of Eqs. 2, 3, 4 and 5). For the MOS score estimation, the codec used must be considered. For simplification, Fig. 5 shows only the mapping rule considering the G.711 codec and an end-to-end delay greater than 177.3ms.

```

1. CoS(?cos)∧
2. hasParameter(?cos,?owdparam)∧OneWayDelay(?owdparam)∧
3. hasParameter(?cos,?plrparam)∧PacketLossRate(?plrparam)∧
4. hasMeasure(?owdparam,?owdmeasure)∧
5. qualityLiteralValue(?owdmeasure,?d)∧
6. hasMeasure(?plrparam,?plrmeasure)∧
7. qualityLiteralValue(?plrmeasure,?p)∧
8. hasEquivalent(?cos,?mosspec)∧hasParameter(?mosspec,?mos)∧
9. MeanOpinionScore(?mos)∧
10. swrlb:add(?de2e,?d,80)∧
11. swrlb:greaterThanOrEqual(?de2e,177.3)∧
12. swrlb:add(?var1,?p,4.3)∧swrlb:divide(?var2,?p,?var1)∧
13. swrlb:multiply(?ieff,95,?var2)∧
14. swrlb:subtract(?var3,?de2e,177.3)∧
15. swrlb:multiply(?var4,?var3,0.11)∧
16. swrlb:multiply(?var5,?de2e,0.024)∧swrlb:add(?id,?var5,?var4)∧
17. swrlb:subtract(?var6,93.2,?id)∧swrlb:subtract(?r,?var6,?ieff)∧
18. swrlb:subtract(?var7,100,?r)∧swrlb:subtract(?var8,?r,60)∧
19. swrlb:multiply(?var9,?var8,?var7)∧
20. swrlb:multiply(?var10,?var9,?r)∧
21. swrlb:multiply(?var11,?var10,7.0E-6)∧
22. swrlb:multiply(?var12,0.035,?r)∧
23. swrlb:add(?var13,?var12,?var11)∧
24. swrlb:add(?mosvalue,1,?var13)∧
25. qualityLiteralValue(result,?mosvalue)
26. ⇒ hasMeasure(?mos,result)

```

Fig. 5 MOS-mapping rules

All SWRL rules are expressed in terms of OWL concepts (classes, properties, individuals). The SWRL rules have the form of an implication between an antecedent (body) and a consequent (head). As defined in [13], an SWRL can be denoted as $R = \text{Implies}(\text{Antecedent}(A), \text{Consequent}(C))$. The intended meaning is: whenever the conditions specified in the antecedent (A) hold, then the conditions specified in the consequent (C) must also hold. Both the antecedent and consequent consist of zero or more SWRL atoms. SWRL atoms can be of the form $C(x)$, $P(x,y)$, $\text{sameAs}(x,y)$, $\text{differentFrom}(x,y)$, or $\text{builtIn}(r,x,...)$, where C is an OWL description or data range, P is an OWL property, r is a built-in relation, x and y are either variables, OWL individuals or OWL data values. SWRL built-ins are used to perform specific mathematical computation (*swrlb:add*, *swrlb:divide*, *swrlb:multiply*, *swrlb:pow*) and comparisons (e.g., *swrlb:equal*).

In Fig. 5, we adopt the informal “human readable” form to present the MOS mapping rules (as proposed by [13]). In this syntax, a rule has the form: $\text{Antecedent}(A) \Rightarrow \text{Consequent}(C)$, where both antecedent and consequent are conjunctions of atoms written as $a1 \wedge \dots \wedge an$. Variables are indicated using the standard convention of prefixing them with a question mark (e.g., *?argument*).

In the rule presented in Fig. 5, the conjunction of atoms in lines 1 to 3 hold if there is an individual *CoS* containing a QoS specification using the OWD and PLR parameters (therefore, allowing its mapping to a MOS score). Lines 4 to 7 contain a conjunction of atoms allowing the storing of OWD and PLR values in variables *d* and *p*, respectively. Next, the atoms in lines 8 and 9 allow the identification of the individual *mos* of the class *MeanOpinionScore* that will store the MOS score to be estimated by this rule. This individual is the QoS parameter used by *QoSSpec* *mosspec* that in its turn is related to the evaluated *CoS* individual by the *hasEquivalent* property. Line 10 allows the calculation of the end-to-end delay using a typical buffering delay for G.711. The atom in Lines 11 holds if this delay is greater than 177.3 ms (see Eq. 3).

Lines 12 to 18 represent Eqs. 2, 3 and 4, allowing the estimation of the value of *R*. To describe these in SWRL, it was necessary to use auxiliary variables holding intermediate values of the various mathematical computations and comparisons present in this equation. In Fig. 5 these auxiliary variables are indicated by *?varn*. For instance, the variable *var1* holds the product of *?p* and 4.3. Lines 19 to 24 represent Eq. 5 used to estimate the MOS score from the *R* value. Finally, the atoms in lines 25 and 26 allow the storing of the estimated MOS score as a measure of the individual *mos*.

5.3 The inference process

As presented in Sect. 4.2, using NetQoSOnt the NSP can verify that network service satisfies the user requirement using the class hierarchy generated by the inference process. If the class specifying the user requirement is inferred as a subclass of *QoSSpec* specifying a quality guaranteed by a service provided by this NSP, then this service meets the user requirement. The rationale is different in the proposed NetQoSOnt extension. As presented in the previous section, now the quality level guaranteed by the network service is specified by an individual and the user requirement is specified by a subclass of *QoSSpec*.

The classification is one of the description logic inference services, which computes the subclass relations between every named class to create the complete class hierarchy. Another service is the realization [44], which finds the most specific classes that an individual belongs to. The realization is used here to verify that there is an individual *CoS* that belongs to the subclass of *QoSSpec* specifying the user requirements. If yes, the *CoS* represented by the individual belonging to the subclass *QoSSpec* specifying, therefore, that the user requirements meets the user requirements.

6 An illustrative scenario

This section illustrates how the proposed NetQoSOnt extension can be used to address the problem of QoE/QoS mapping

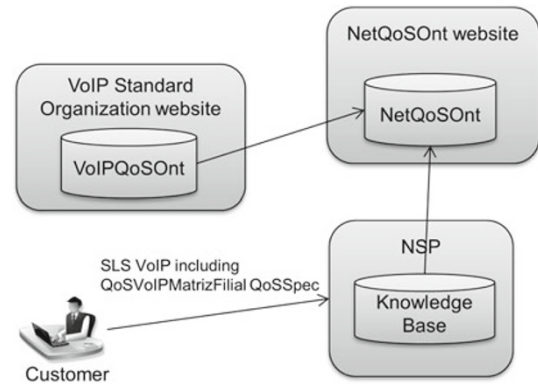


Fig. 6 Illustrative scenario

during a QoS service negotiation. Moreover, this illustrative scenario was used as a testing scenario.

In this illustrative scenario, presented in Fig. 6, we first consider that NetQoSOnt was used by a hypothetical standards organization to describe a set of QoS/QoE-related concepts in the VoIP domain, and the resulting file was published in <http://VoIPOrg.org/VoIPQoEOnt.owl>. Some concepts and mapping rules described in VoIPQoSOnt were presented in Fig. 4. Moreover, consider that the NSP adopts the two CoS described in Sect. 4.1.3 and whose QoS specifications are maintained in its KB.

Consider now that a customer negotiates a VoIP SLS with its NSP. This VoIP SLS specifies the voice quality desired by the voice calls between the main office and the branch of its corporation. For this, the customer specifies quality $MOS \geq 4$ for voice calls using the G.711 codec (i.e., he or she accepts voice calls with a quality level inferior to that provided by G.711 in a scenario without loss). The problem stated here is: how can the NSP automatically identify what $MOS \geq 4$ quality means in terms of network QoS parameters and identify which of its CoSs provides the required quality? The solution, using the proposed NetQoSOnt extension, is described in the following subsections.

To simulate and test our own proposed NetQoSOnt extension, an SLS negotiation system prototype has been implemented using Java, API OWL [45], and the OWL reasoner Pellet [44]. This is a proof-of-concept prototype, implementing the minimum functionalities required for the customer of a VoIP service to specify the required quality level and for the NSP to reach the CoS that meets the customer requirements. Figure 7 presents the two elements of the implemented SLS Negotiation System:

- *QoS Specification Creator*: it is a Java application built to emulate the client side of our SLS Negotiation System. This application offers an interface, presented in Fig. 8, through which a customer can request a voice quality by defining a MOS score rate, an *R* factor rate or using

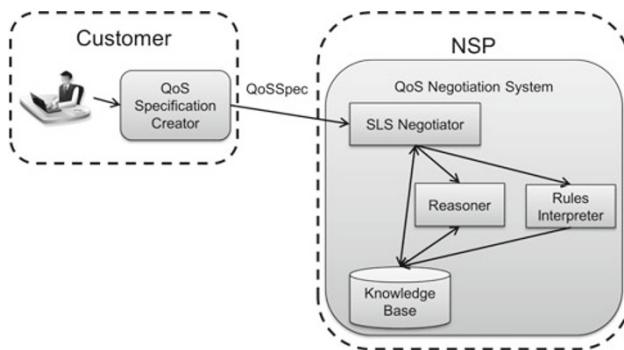


Fig. 7 QoS negotiation system prototype

a subjective quality. The mappings between these different QoE parameters are presented in Table 1. Based on VoIPQoS Ont, the QoS Specification Creator generates the OWL classes and individuals describing the QoS specification required by the customer.

- *QoS Negotiation Systems*: it is a Java application responsible for receiving a QoS specification, a subclass of *QoS Spec* expressing the customer requirements and identifying what CoS meets the customer requirements. This system is composed by three components—SLS Negotiator, Rules Interpreter, and Knowledge Base (KB).

During the QoS negotiation, the SLS Negotiation System proceeds as follows. The SLS Negotiator receives the subclass of *QoS Spec* and verifies if there is already in the KB an equivalent subclass of *QoS Spec*. If yes, then the inference process has already been done, and the SLS Negotiator can reach the CoS satisfying the quality requirement of the customer and present it to the user's application (as presented in Sect. 5.3). Otherwise, the new *QoS Spec* must be included in the KB. The Rules Interpreter is called when new QoE/QoS parameters are used by a subclass of *QoS Spec* specifying the customer's requirements. Before the rules are applied, this component is responsible for including in the KB new individuals belonging to the classes *CoS* and *Parameter* using the new QoE/QoS parameter for each individual *CoS*. After this operation, the SLS Negotiator can call the Reasoner to do the classification and realization on the KB. Finally, the SLS Negotiator can reach the CoS satisfying the quality requirement of the customer and present it to the user's application.

The SWRL language does not support the dynamic generation of OWL individuals. This automatic generation of individuals belonging to the class *QoS Spec* (and other related individuals) based on the SWRL mapping rules is accomplished by the Rules Interpreter in our QoS Negotiation System. With this objective, we developed a simple SWRL interpreter, implemented in Java, and used the OWL API to populate the KB of the NSP with the new individuals.

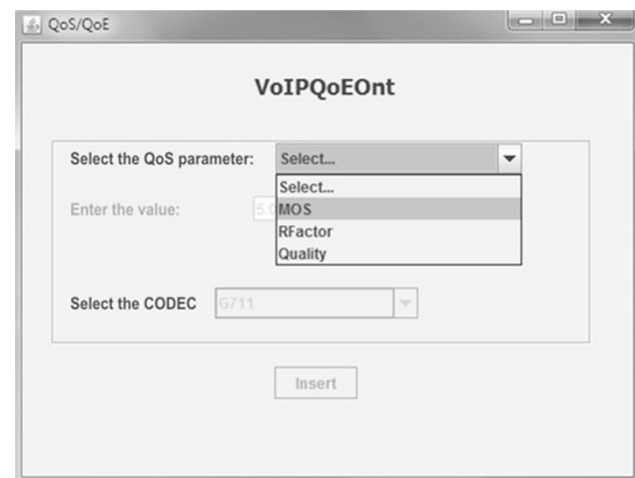


Fig. 8 Customer interface of the QoS specification creator

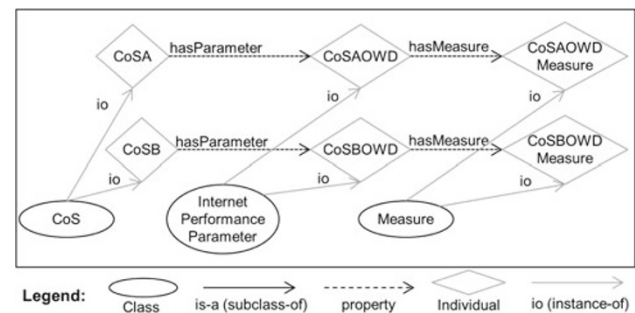


Fig. 9 QoS specification guaranteed by the CoSs as maintained in the KB

6.1 Specifying QoS-related concepts

The NSP must include in its KB a set of concepts specifying the QoS level guaranteed by its CoS. These concepts are presented in Fig. 9. For simplification, this figure presents only OWD metrics; however, other performance metrics were considered, such as IPDV and PLR. Figure 9 illustrates two individuals belonging to the class *CoS*: *CoSA* and *CoSB*. *CoSA* is a subclass of *QoS Spec* specifying the QoS level guaranteed by this CoS. *CoSA* has relations with *CoSAOWD* and *CoSAPLR*, representing the maximum OWD and PLR guaranteed by *CoSA*, respectively. The measure of the maximum OWD is defined by *CoSAOWD Measure* as ≤ 140 ms, and the measure of the maximum PLR is defined by *CoSAPLR* as ≤ 0.001 . In its turn, *CoSB* has relations with *CoSBOWD* and *CoSBPLR* specifying the maximum OWD and PLR guaranteed by this CoS. The measures of the maximum OWD and PLR are ≤ 400 ms and ≤ 0.001 , respectively. In our QoS Negotiation System prototype, these concepts have been manually included in the KB.

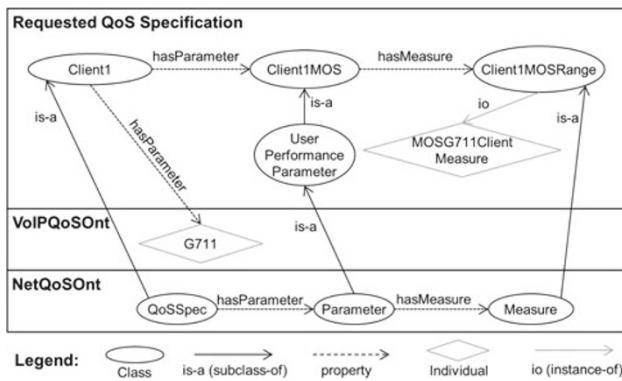


Fig. 10 Required QoS specification

6.2 Specifying the required quality

The classes and individuals generated during the service negotiation will be maintained in the KB of the NSP. Figure 10 presents the classes and individuals describing the QoS specification under negotiation ($MOS \geq 4$ with codec G.711). The QoS specification is described by *Client1*, a subclass of *QoSSpec*. *Client1* has two parameters: *G.711*, an individual of the class *Codec* specifying the codec to be used by the VoIP calls; and *Client1MOS*, a subclass of <http://VoIPOrg.org/VoIPQoEOnt.owl/#MOS>. This subclass has a measure specified by *Client1MOSRate* that in turn has the value *qualityLiteralValue double=4*. This value is stated in the individual *MOSClient1ClientMeasure*. Note that if the class <http://VoIPOrg.org/VoIPQoEOnt.owl/#MOS> is not known, the negotiation system will import this description from <http://VoIPOrg.org/VoIPQoEOnt.owl>. This is so that new QoE and QoS parameters may become known by the NSP.

6.3 Customer's request treatment

During the VoIP SLS negotiation, the SLS Negotiator must accept/renege or refuse the requested SLS. One of the first steps is to verify which CoSs can meet the customer's needs. More specifically, the QoS specification used by the customer must be compared with the QoS specification guaranteed by the CoSs. To allow this comparison, it is necessary to map the QoS specifications of the CoSs into QoS specifications using the same QoE parameters used by the customer. In our illustrative scenario, the NSP must verify what CoSs offer the quality $MOS \geq 4$.

Upon receiving the customer's request, the SLS Negotiator verifies if the QoS specification is already available in the KB. If it is available, the module already knows which CoS meets the user's request. Let's consider here that the NSP does not yet know the MOS concept. The procedure begins with the importation of the concepts used in the

QoS specification, i.e., the importation of <http://VoIPOrg.org/VoIPQoEOnt.owl>.

The property *mappingTo* (Fig. 4) allows the SLS Negotiator to identify the individual containing the mapping rules of the QoE parameter used by the customer (i.e., MOS). In this illustrative scenario, *mappingTo* identifies the mapping rules of the parameter MOS, which are those kept in the individual *QoSSto-MOS-SWRL* of the class *MOSMapping*. In order to determine the minimum MOS score guaranteed by each CoSs, the SLS Negotiator calls the Rules Interpreter (Fig. 7) to interpret these mapping rules kept in the individual *QoSStoMOS-SWRL*. Interpreting these rules, the Rules Interpreter creates new individuals belonging the class *CoS* specifying the quality level guaranteed by CoSA and CoSB, now using MOS parameters and codec G.711. For this, the mapping rule available in the individual *QoSStoMOS-SWRL* is interpreted twice, each one considering the OWD and PLR guaranteed by one of the CoSs of the NSP. The output variable's value returning from each interpretation is the minimum MOS score guaranteed by the considered CoS.

Figure 11 presents the individuals generated from these two interpretations: the individuals *QoSSpecMOS-G711CoSA* and *QoSSpecMOSG711CoSB* represent the QoS specifications using the MOS parameter and the G.711 codec guaranteed by CoSA and CoSB, respectively. The individuals *MOSG711CoSAMeasure* (double = 4.1) and *MOSG711CoSBMeasure* (double = 2.96) specify the minimum MOS score guaranteed by CoSA and CoSB, respectively.

After generating the individuals belonging to the class *QoSSpec*, the QoS negotiation system must use the Reasoner

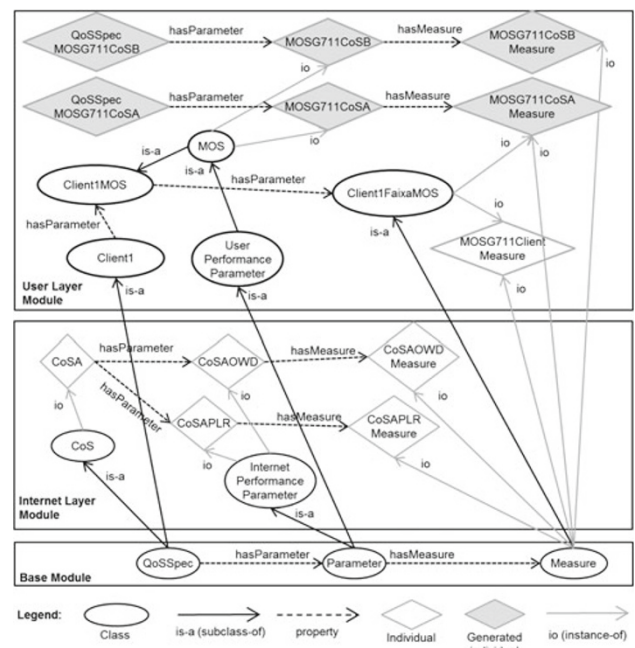


Fig. 11 Generated individuals

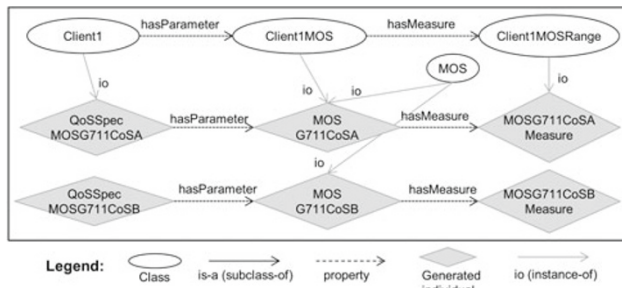


Fig. 12 Generated individuals and the inference results

to identify what CoSs ensure the quality requested by the client. As *Client1* is the formalization of the requested quality level, the problem of identifying what CoS ensures this quality level can be represented as the identification of which individuals belonging to the class *CoS* also belong to the class *Client1*. The new hierarchy obtained after the inference process is presented in Fig. 12, where one can observe that *QoSSpecMOSG711CoSA* is an individual of the class *Client1*, and therefore, CoSA meets the quality requested by the client.

To identify individuals belonging to subclasses of *QoSSpec*, the Reasoner analyzes the intervals defined by the *Measure* subclasses and individuals. Since $4.1 \geq 4$, the reasoner will infer that the individual *MOSG711-CoSAMeasure* belongs to the class *Client1MOSRate*. Consequently, *MOSG711CoSA* will be inferred as an individual belonging to the class *Client1MOS* and *QoSSpecMOS-G711CoSA* as individual belonging to the *Client1*. This makes us conclude that CoSA offers a QoS level that meets the customer's request.

This hypothetical scenario illustrates the flexibility offered by our NetQoSOnt extension: the NSP, without knowing the QoE parameter chosen by the user, can dynamically generate individuals in the user layer of its KB. These instances were generated based on the mapping rules published by a hypothetical VoIP standards organization. These individuals were used to identify what CoS meets the user requirement. These operations are necessary only when the NSP does not know the QoE parameter used. If the NSP knows the QoE parameter used, it also knows the quality level provided by their CoSs in terms of these QoE parameters.

In order to test the effectiveness of our semantic-based approach, various negotiation scenarios were simulated by altering the QoE parameter and value (considering the maximum values for G.711): MOS scores from 1.0 to 4.1; R values from 1 to 82; and all subjective qualities (High, Medium, Low and Poor). In all simulations the CoS automatically identified as meeting the customer's request is the same than those manually found using the same rules.

6.4 Performance evaluation

For this illustrative scenario, the inference process took 312 ms on a standard computer with Intel i5 and 8 GB RAM. In a real situation, the time of transfer of the new quality parameter and their mapping rules (MOS scenario illustrates) must be added to this delay. Considering that the small sized ontologies described do reflect a real provider specification (in general, an NSP has only a small set of CoSs), the time response of the reasoner proves viable the use of ontologies to help negotiation process. It is important note that this transfer and mapping procedure is necessary only the first time that QoE/QoS parameter is used.

6.5 Applying the proposal in other domains

The semantic approach for QoS/QoE mapping proposed in this paper can be applied in other domains. For instance, using the QoS/QoE correlation model proposed by [46], it is possible to specify QoE metrics for IPTV services, including their mapping rules. For that purpose, it is necessary to specify the QoE concepts in the IPTV domain (analogous to the specification of the QoE concepts in the VoIP domain presented in Fig. 4) and publish it in a public web site.

7 Conclusions

This paper proposed an extension of the NetQoSOnt ontology offering resources to automatically map QoS specifications in the different network layers. Using the SWRL language to specify mapping rules between QoE/QoS parameters brings flexibility to network service providers, application implementers, and standards organizations: new QoS/QoE parameters and mapping rules can be published and be automatically interpreted by service providers.

Our proposal was illustrated and tested using a QoS service negotiation prototype. Using this prototype, we showed the effectiveness of our semantic-based approach when used during the VoIP service negotiation. In this scenario, the customer can adopt different QoE parameters and the Network Service Provider can identify what network service satisfies the customer's requirement.

Our approach adopts the SWRL language to specify the mapping rules. However, one rule specified in SWRL containing a high number of characteristics will make the rule too long and difficult to understand and manage, even with a simple formula [47]. To overcome these disadvantages, [47] combines SWRL rules and OpenMath [34]: SWRL is used to manage the OWL instances by locating those required in the formulas and OpenMath defines the mathematical

functions. This solution can be used in our approach, changing the way the mapping rules are specified. However, this solution requires the use of SWRL extensions not yet standardized and there are no tools supporting it.

As future works, we are defining a complete semantic-based solution for network service negotiation. Additionally, we are also populating our ontology with new QoE parameters and the definition of new mapping rules using the SWRL language.

References

- Kilkki K (2008) Quality of experience in communications ecosystem. *J Univers Comput Sci* 14(5):615–624
- Moorse A (2001) Metrics for the internet age: quality of experience and quality of business. In: *Proceedings of the fifth international workshop on performativity modeling of computer and communication systems*, pp 26–31
- ITU-T Recommendation P.800 (2003) Methods for subjective determination of transmission quality
- Siller M, Woods J (2006) Using an agent based platform to map quality of service to experience in conventional and active networks. *IEE Proc Commun* 153(6):828–840
- Ghinea G, Thomas J (1999) Quality of perception to quality of service mapping using a dynamically reconfigurable communication system. In: *Proceedings of the IEEE global telecommunications conference*, vol 4, pp 2061–2065
- Hung PCK, Li H, Jeng JJ (2005) WS-negotiation: an overview of research issues. In: *Proceedings of the 37th annual Hawaii international conference on system sciences*, 2004, pp 33–42
- Davies J, Studer R, Warren P (2006) *Semantic web technologies trends and research in ontology-based systems*. Wiley, London
- W3C (2004) The overview of OWL web ontology language. <http://www.w3.org/TR/owl-features/>. Accessed Aug 2012
- Rodrigues C, Carvalho P, Luis MAS, Solange RL (2012) An ontology for managing network services quality. *Exp Syst Appl: Int J* 39(9):7938–7946
- Zrelli S, Ishida A, Okabe N, Teraoka F (2012) ENM: a service oriented architecture for ontology-driven network management in heterogeneous network infrastructures. In: *Proceeding of the IEEE/IFIP 4th workshop on Management of the Future Internet (ManFI)*, pp 1096–1103
- Hachem S, Teixeira T, Issarny V (2011) Ontologies for the Internet of things. In: *Proceedings of the 8th Middleware doctoral symposium*, Article No 3
- Prudêncio AC, Willrich R, Tazi S, Diaz M (2009) Quality of service specifications: a semantic approach. In: *Proceedings of the 8th IEEE international symposium on network computing and application*, pp 219–226
- W3C (2004) SWRL: a semantic web rule language combining OWL and RuleML. <http://www.w3.org/Submission/SWRL/>. Accessed 22 Feb 2012
- ITU-T Recommendation E.800 (1994) Terms and definitions related to quality of service and network performance including dependability
- Goderis D et al. (2003) Service level specification semantics and parameters. <http://tools.ietf.org/html/draft-tequila-sls-02>. Accessed Aug 2012
- ITU-T Recommendation Y.1541 (2006) Network performance objectives for IP-based services
- ITU-T Rec. G.100/P.10 (2007) Amendment 1: new appendix I definition of Quality of Experience (QoE)
- ITU-T P.910 (1999) Subjective video quality assessment methods for multimedia applications
- Wang Y (2006) Survey of objective video quality measurements. Technical report, Worcester Polytechnic Institute
- ITU-R Rec. BT.500-11 (2003) Methodology for the subjective assessment of the quality of television pictures
- ITU-T Rec. G.107 (2011) G.107 The E-Model, a computational model for use in transmission planning
- ITU-T Rec. P.862 (2001) Perceptual evaluation of speech quality (PESQ): an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs
- ITU-T Rec. G.113 (2007) Transmission impairments due to speech processing
- Cole R, Rosenbluth J (2001) Voice over IP performance monitoring. *ACM SIGCOMM Comput Commun Rev* 31(2):9–24
- Fujimoto K, Ata S, Murata M (2002) Adaptive playout buffer algorithm for enhancing perceived quality of streaming applications. In: *Proceedings of the IEEE global telecommunications conference*, vol 1, pp 2463–2469
- Zhou C, Chia L, Lee B (2005) QoS measurement issues with DAML-QoS ontology. In: *Proceedings of the IEEE international conference on E-Business engineering*, pp 395–403
- Dobson G, Lock R, Sommerville I (2005) Quality of service requirements specification using an ontology. In: *Workshop on service-oriented computing requirements (SOCCER)*
- Bleul S, Weise T (2005) An ontology for quality-aware service discovery. In: *Proceedings of the first international workshop on engineering service compositions*, pp 35–42
- Kritikos K, Plexouakis D (2007) OWL-Q for semantic QoS-based web service description and discovery. In: *Proceedings of the first international joint workshop on service matchmaking and resource retrieval in the semantic web*, pp 123–137
- Gallo E, Siller M, Woods J (2007) An ontology for the quality of experience framework. In: *Proceedings of the IEEE international conference on systems, man and cybernetics*, pp 1540–1544
- Moraes P, Sampaio L, Monteiro J, Portnoi M (2008) MonONTO: a domain ontology for network monitoring and recommendation for advanced internet applications users. In: *Proceedings of the IEEE network operations and management symposium workshops*, pp 116–123
- Royer JC, Willrich R, Diaz M (2008) User profile-based authorization policies for network QoS services. In: *Proceedings of the IEEE international symposium on network computing and applications*, pp 68–75
- Macian AS, Lopez D, Lopez de Vergara JE, Pastor E (2006) A framework for the automatic calculation of quality of experience in telematic services. In: *Proceedings of the 13th HP Open View University Association Workshop*
- OpenMath (2012) <http://www.openmath.org>. Accessed Aug 2012
- Green L (2006) Service level agreements: an ontological approach. In: *Proceedings of the 8th international conference on electronic commerce*, pp 185–194
- Chen G, Bai X, Huang X, Li M, Zhou L (2011) Evaluating services on the cloud using ontology QoS model. In: *Proceedings of the IEEE 6th international symposium on service oriented system, engineering*, pp 312–317
- Fallon L, O'Sullivan D (2012) Using a semantic knowledge base for communication service quality management in home area networks. In: *IEEE network operations and management symposium (NOMS)*, pp 43–51
- Alípio P, Neves J, Carvalho P (2007) An ontology for network services. *Comput Inf* 26(5):543–561
- Berrueta D, Polo L (2008) Measurement units ontology working draft. <http://idi.fundacionctic.org/muo/muo-vocab.html>. Accessed Aug 2012

40. Protégé (2012) The Protégé ontology editor and knowledge acquisition system. <http://protege.stanford.edu/>. Accessed Aug 2012
41. Exposito E, Diaz M, Sénac P (2004) Design principles of a QoS-oriented transport protocol. In: Proceedings of the international conference on intelligence in communication systems, pp 151–159
42. Willrich R, Vicente LH, Uriarte RB, Prudêncio AC, Cé Júnior J (2009) Dynamic invocation of services with QoS guarantees in SIP multimedia sessions (in Portuguese). In: Proceedings of the 8th international information and telecommunication technologies, symposium (I2TS)
43. Schulzrinne H, Casner S (2003) RFC 3551: RTP profile for audio and video conferences with minimal control
44. Sirin E et al (2007) Pellet: the open source OWL DL reasoner. Web Semantics: Science, Services and Agents on the World Wide Web 5(2):51–52
45. Owl API (2012) <http://owlapi.sourceforge.net/>. Accessed Aug 2012
46. Kim HJ, Choi SG (2010) A study on a QoS/QoE correlation model for QoE evaluation on IPTV Service. In: Proceedings of the 12th international conference on advanced communication technology, vol 2, pp 1377–1382
47. Macián AS, Pastor E, Vergara JEL, López D (2007) Extending SWRL to enhance mathematical support. In: Proceedings of the 1st international conference on web reasoning and rule systems, pp 358–360