A Model-driven WSDL Extension for Describing the QoS of Web Services*

Andrea D'Ambrogio
Dept. of Computer Science Systems and Production
University of Roma TorVergata
E-Mail: dambro@info.uniroma2.it

Abstract

Web services are the building blocks of the emerging computing paradigm based on service-oriented architectures. A web service is a self-describing, open component that supports rapid composition of distributed applications. Web service definitions are used to describe the service capabilities in terms of the operations of the service and the input and output messages for each operation. Such definitions are expressed in XML by use of the Web Service Definition Language (WSDL). Unfortunately, a WSDL description only addresses the functional aspects of a web service without containing any useful description of non-functional or quality of service (QoS) characteristics. This paper introduces a lightweight WSDL extension for the description of QoS characteristics of a web service. The extension is carried out as a metamodel transformation, according to principles and standards recommended by the Model Driven Architecture (MDA). The WSDL metamodel is introduced and then transformed into the Q-WSDL (QoS-enabled WSDL) metamodel. The proposed Q-WSDL extension can effectively be used to specify QoS requirements, to establish service level agreements (SLA), to add QoS-oriented characteristics when querying registries of web services and to support the automated mapping from WSDL documents to Q-WSDL ones and from UML models to Q-WSDL web services.

1. Introduction

Service-oriented computing is becoming the prominent paradigm for distributed computing and e-commerce. Web services are the building blocks for the application of service-oriented computing on the Web [20].

A web service is a self-describing, open component that supports rapid composition of distributed applications. In a service-oriented architecture, the service provider creates a *WSDL (Web Service Description Language)* service description and publishes it to one or more discovery registries (such as UDDI), so that service consumers can find the service using a wide variety of search criteria and then use the WSDL description to develop or configure a client that will interact with the service.

A WSDL description is an XML document that contains all the information about service capabilities and invocation mechanisms. The capabilities are described in terms of the operations of the service and the input and output messages for each operation. What is needed to invoke the service is provided by a binding implementation description that describes how messages are sent through the network to reach the service location, where the hosting environment executes the service implementation.

Unfortunately, a WSDL document only addresses the functional aspects of a web service without containing any useful description of non-functional or quality of service (QoS) characteristics. Different web services may provide similar functionality, but with distinct quality of service properties. In the selection of a web service, it is important to consider both functional and QoS properties in order to fully satisfy the needs of a service consumer [9, 12].

This paper introduces a lightweight WSDL extension for the description of QoS characteristics of a web service, such as performance, reliability, availability, security, etc.

The proposed extension is defined by first introducing the WSDL metamodel, derived from the WSDL XML Schema, and then transforming it into a *QoS-enabled WSDL (Q-WSDL)* metamodel, from which the Q-WSDL XML Schema is derived. Indeed, an XML Schema defines the WSDL language in the same respect as a metamodel is used to define a model. Representing the WSDL grammar in terms of a metamodel allows to enhance its comprehensibility and facilitate its extension.

The WSDL and Q-WSDL metamodels are defined by use of the *Meta Object Facility (MOF)*, the Object Management Group's (OMG) standard for specifying technology neutral metamodels, or models used to describe other mod-

* Work partially supported by funds from the FIRB project on *Performance Evaluation of Complex Systems* and from the University of Roma TorVergata CERTIA Research Center.

IEEE International Conference on Web Services (ICWS'06)
0-7695-2669-1/06 $20.00 © 2006 IEEE

els [14]. The production of XML Schemas from MOF metamodels, and vice-versa, is specified by a further OMG's standard, the *XML Metadata Interchange (XMI)* specification. MOF and XMI are at the heart of the *Model Driven Architecture (MDA)*, the OMG's effort that focuses on models as the primary artifacts of software development, evolution and integration [13]. Applying MDA to Web services has recently received considerable attention, mainly addressing the automated generation of platform-specific implementations from web service models [1, 6]. In this paper, MDA standards are also used to formally define and apply the QoS-oriented extension of WSDL, in order to appropriately enrich web service descriptions with QoS data.

The Q-WSDL extension can be applied to serve different purposes, such as:

- to specify QoS requirements of web services;
- to add QoS characteristics when querying registries of web services (e.g., UDDI);
- to define service level specifications (SLS) when establishing service level agreements (SLA);
- to enable the QoS-aware composition of web services;
- to ease the derivation of models that predict the QoS of web services;
- to support the automated mapping from WSDL documents to Q-WSDL ones;
- to support the automated mapping from UML models (e.g., UML models of BPEL processes) to services in Q-WSDL.

The paper is organized as follows. Section 2 overviews related contributions. Section 3 describes the main steps of the extension process and clarifies the terminology used throughout the paper. Section 4 introduces the characteristics used to describe the QoS of web services, while Section 5 illustrates the proposed Q-WSDL metamodel. Finally, Section 6 gives a Q-WSDL example application.

2. Related Work

Aspects that focus on describing, advertising and signing up to web services at defined levels of quality of service have also been addressed by similar and bigger efforts [22], such as the HPs Web Services Management Framework (WSMF) [4], the IBMs Web Service Level Agreement (WSLA) language [8], the Web Services Offer Language (WSOL) [23] and the WS-Policy [3]. Some contributions have also been given in the Semantic Web area (see OWL-S [11] and METEOR-S [24]).

Such considerable contributions introduce new infrastructures for specifying the QoS characteristics of web services and specifically target web service management activities. This paper is instead in favor of an approach that extends the WSDL capabilities rather than introducing additional languages on top of it.

The proposed approach is *lightweight*, so that existing WSDL documents can easily be extended without altering their original content, and *automated*, in order to effectively support its adoption. Moreover, Q-WSDL descriptions of web services can be used to seamlessly derive predictive models that assist service providers to effectively managing the levels of QoS characteristics. The Q-WSDL metamodel is an important contribution in such a direction because methods for the automated building and evaluation of predictive models based on metamodel transformations are currently being developed (see, e.g., [5] and [21]) and the Q-WSDL metamodel includes the formal description of QoS parameters required by such methods. Service providers can thus be endowed with methods and tools to obtain QoS predictions and answer questions like: "what is the expected service quality?" or "what is a satisfactory reconfiguration of the service parameters to guarantee an agreed QoS?" or also "what is the acceptable response time under different workload situations?".

A further advantage of having defined the extension at metamodel level is that it is relatively easy to foresee automated transformations of both WSDL documents into Q-WSDL ones and platform-independent UML models (PIMs in MDA terms) into platform-specific web services in Q-WSDL (PSMs in MDA terms). The degree of automation will further be increased by the availability of tools implementing the MOF 2.0 Query/View/Transformation (QVT) specification, which has just been adopted by the OMG [15].

3. Extension Process

As said in Section 1, the proposed extension has been carried out as a metamodel transformation, according to principles and standards recommended by MDA.

The set of guidelines provided by MDA strongly relies on metamodeling techniques for structuring specifications expressed as models and transformations between such models [13]. In this respect, MOF is the key standard that provides an abstract language and a framework for specifying, constructing, and managing technology neutral metamodels, or models used to describe other models [14]. In MDA terms, a *model* (e.g., an UML model [18]) is an instance of a *MOF metamodel* (e.g., the UML metamodel [18]), which in turn is an instance of the MOF *meta-metamodel* specified in [14] (and briefly called *MOF Model*).

A side standard of MOF is the OMG's XMI specification, which provides a set of rules to serialize models and MOF metamodels into *XML documents* and *XML Schemas*, respectively, and to derive a MOF metamodel from an XML Schema [26, 19].

Figure 1 illustrates the metamodel-based process used to carry out the QoS-oriented extension of WSDL, with the relationships among the main elements in the process.

A WSDL description is basically an XML document that is produced and validated by use of the WSDL XML Schema described in [25]. The relationship between an XML document and its corresponding XML Schema is the same that links a model and its corresponding metamodel. Indeed, an XML Schema defines the WSDL language in the same respect where a metamodel is used to define a model. Restated as an `<<instance of>>` relationship, an XML document is an instance of an XML Schema in the same way a model is an instance of a metamodel.

Representing the WSDL grammar in terms of a metamodel allows to enhance its comprehensibility and facilitate its extension. A MOF metamodel is indeed expressed by use of well-known visual modeling constructs such as classes, attributes and associations, which are easy to understand and analyze compared with the verbosity and poor readability of XML Schema structures. Hence, as a first step in the process, the WSDL metamodel has been derived from the WSDL XML Schema. Such a step is based on XMI rules and can be fully automated, as described in [2].

As a second step, the WSDL metamodel has been extended by applying a metamodel transformation that maps the elements of a source metamodel (i.e., the WSDL metamodel) to elements of a target metamodel (i.e., the Q-WSDL metamodel). The transformation is inspired by the following UML profiles published by the OMG:

- the UML Profile for Quality of Service and Fault Tolerance (hereafter *QoS Profile*), which extends the UML metamodel to provide UML models with QoS-oriented annotations [16];
- the UML Profile for Schedulability, Performance and Time (hereafter *SPT Profile*), which extends the UML metamodel to provide UML models with time and performance-oriented annotations [17].

Concepts stemmed from the QoS and SPT profiles are here used to extend the WSDL metamodel and thus obtain the Q-WSDL (QoS-enabled WSDL) metamodel, so that WSDL models (and their respective XML documents) can effectively be enriched with QoS data.

Finally, as a third step of the extension process, the Q-WSDL metamodel has been serialized to the Q-WSDL XML Schema by use of XMI-based rules. The obtained Q-WSDL Schema is then used to produce and validate Q-WSDL XML documents, i.e. QoS-enabled WSDL descriptions.

4. QoS characteristics of Web Services

QoS can be defined in terms of the characteristics that contribute to the overall quality of a service as perceived by service consumers [7].

In the QoS Profile, a *QoS characteristic* is a quantified aspect of the QoS, for example latency, throughput, reliability, availability, etc., which is defined independently of the means by which it is represented, managed or controlled. QoS Characteristics of a common subject are grouped into abstract *QoS categories*, for example performance (for latency and throughput characteristics) and dependability (for reliability and availability characteristics). A QoS characteristic is quantified by use of *QoS dimensions*. Examples dimensions for reliability are MTBF (mean time between failures), MTTR (mean time to repair), number of failures supported, etc. [16].

The QoS profile introduces a catalog of general QoS characteristics within a structured collection of modeling concepts, denoted as QoS modeling framework. Different domains usually requires different catalogs of QoS characteristics, and thus the QoS modeling framework provides enough flexibility to either specialize the QoS characteristics for a given application domain or introduce new characteristics.

In this respect, this section introduces a catalog of QoS characteristics of interest in the web services domain. Some QoS characteristics of the QoS Profile catalog have been reused and some new characteristics have been defined.

The catalog of characteristics and dimensions that describe the QoS of web services consists of the following classes:

- `OperationLatency`: time taken to process a single operation, quantified in terms of:
  - `ServiceTime`: time to execute the operation at server side (does not include network transfer time);
  - `TurnAround`: end-to-end time to complete an operation, as experienced by a service consumer (including the time to transfer requests and responses from and to the service consumer).
- `OperationDemand`: intensity of the demand for the operation, quantified in terms of:
  - `ArrivalRate`: rate at which service consumers request the operation;
- `Network`: characteristics of the IP network to which the service's hosting environment is connected, quantified in terms of:
  - `BitRate`: rate at which the IP network transfers user data;

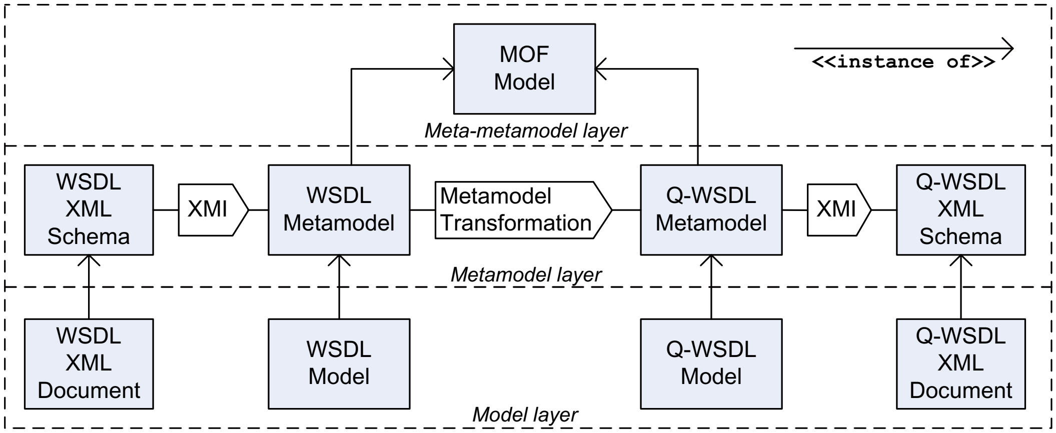

Figure 1. WSDL extension as a metamodel transformation

  - `Delay`: end-to-end delay experienced by IP packets while passing through the network;
  - `Jitter`: variations in the end-to-end delay;
  - `PacketLoss`: ratio of the number of undelivered packets to sent ones.
- `ServiceAccessControl`: protection against unauthorized access to the service, quantified in terms of:
  - `Policy`: control policy used in the access to the service (e.g., a reference to a WS-Policy document);
- `MessageEncryption`: security mechanisms applied to input and output messages, quantified in terms of:
  - `EncryptionProtocol`: protocol used for securing input and output messages;
  - `KeyType`: type of keys used for message encryption.
- `ServiceAvailability`: readiness of usage of the web service, quantified in terms of:
  - `TimeToRepair`: time-to-repair (TTR) , or time needed to repair and restore service after a failure;
  - `TimeBetweenFailure`: time-between-failure (TBF), or time the application runs before failing;
  - `ExpAvailability`: expected availability, which can be obtained either from expected values of uptime and downtime, as uptime/(uptime+downtime), or from TTR and TBF, as TBF/(TTR+TBF).
- `ServiceReliability`: ability of the web service to keep operating over time, quantified in terms of:
  - `TimeBetweenFailure`: time-between-failure (TBF), or time the service runs before failing;
  - `ExpFailures`: expected number of failures over a time interval, which can be obtained from TBF, as described in [10].

It is worth to note that the main goal is not to provide an exhaustive catalog for the description of the QoS of web service, but to illustrate how the QoS characteristics defined in the QoS Profile's catalog can be customized to be applied to a given application domain. Additional characteristics about, e.g., service utilization, integrity, scalability or accuracy, can be defined or existing ones be modified or removed in a similar way.

5. QoS-enabled WSDL Metamodel (Q-WSDL)

A WSDL document consists of a set of definitions that describe what a service does (the operation it provides), how a service is acceded (data formats and protocols) and where it is located (network address). Such definitions are specified in the WSDL XML Schema [25], which has been used to identify the classes and associations of the WSDL metamodel, illustrated in the portion of Figure 2 bounded by a dashed line shape[1].

Classes and associations outside the dashed line shape in Figure 2 extend the WSDL metamodel to include the description of the QoS characteristics of a web service. The complete set of classes and association in Figure 2 (both

1 For the sake of brevity and readability, only relevant classes are illustrated in Figure 2. Specialized classes (e.g., classes for operations of type *one-way*, *request-response*, *solicit-response* and *notification-response*) have been omitted, as well as classes related to specific documenting and extensibility features of XML.

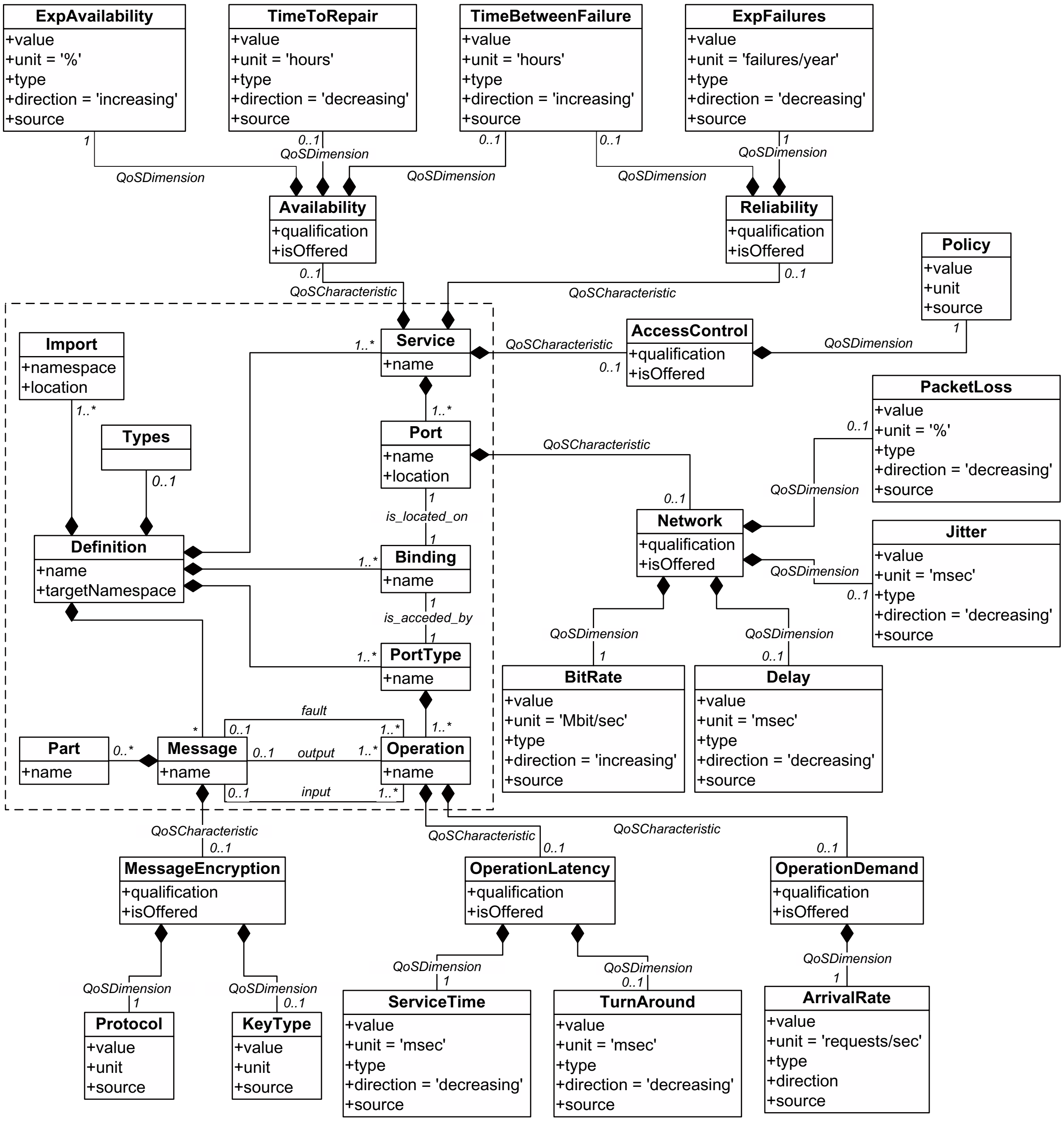

Figure 2. QoS-enabled WSDL (Q-WSDL) metamodel

inside and outside the dashed line shape) identifies the Q-WSDL metamodel.

As said in Section 3, the extension is inspired by the QoS and SPT Profiles, which provide mechanisms for annotating UML models with the description of QoS and time characteristics. The power and flexibility of the QoS Profile has been combined with the simplicity and convenience of expression of the SPT Profile, in order to reach the right balance between flexibility and expressivity.

Each class corresponding to a QoS characteristic provides two attributes:

- `qualification`, which specifies a constraint on the characteristic and can be *guaranteed*, *best-effort*, *threshold best-effort* (i.e., the service issues a warning if the QoS is below the threshold level), *compulsory* *best-effort* (i.e., the system aborts if the QoS is below the compulsory level) or *none*;

- isOffered, whose boolean value specifies if the characteristic is offered (*true*) or required (*false*).

Each class corresponding to a QoS dimension provides the following attributes:

- value, which specifies the value of the dimension;
- unit, which specifies the unit of measure of the dimension;
- source, which specifies how the value has been obtained (can be *required*, *assumed*, *predicted*, *measured* or *undefined*);

When appropriate, a class corresponding to a QoS dimension may also provide the following attributes:

- type, which defines the type of statistical value (can be *maximum*, *minimum*, *mean*, *variance*, *standard deviation*, *i-th percentile* or *distribution*);
- direction, which defines the type of order relation used to compare values (can be *increasing*, *decreasing* or *undefined*);

As usual, associations with multiplicities 0..1 or 0..* denote optional associations, while multiplicities 1 or 1..* denote required associations. This means, for example, that the OperationLatency characteristic must be quantified in terms of the ServiceTime dimension, while the TurnAround dimension is optional (see Figure 2). Attributes that are given values are automatically set to such values when class instances are created in a Q-WSDL model.

The Q-WSDL metamodel in Figure 2 has then been translated into the XML Schema used to produce and validate Q-WSDL XML documents (see Section 3). As an example, the portion of Q-WSDL schema that defines the operation element is as follows[2]:

```
<element name="operation" type="qwsdl:operationType"/>
<complexType name="operationType">
  <complexContent>
   <extension base="qwsdl:documented">
    <choice>
     <group ref="qwsdl:one-way-operation"/>
     <group ref="qwsdl:request-response-operation"/>
     <group ref="qwsdl:solicit-response-operation"/>
     <group ref="qwsdl:notification-operation"/>
     </choice>
     <sequence>
       <element ref="qwsdl:OperationLatency"
          minOccurs="0" maxOccurs="1"/>
     </sequence>
       <attribute name="name" type="NCName"
         use="required"/>
     </extension>
   </complexContent>
</complexType>
<element name="OperationLatency"
  type="qwsdl:OperationLatencyType"/>
<complexType name="OperationLatencyType">
  <sequence>
    <element name="ServiceTime"
      type="qwsdl:ServiceTimeType"
      minOccurs="1" maxOccurs="1"/>
    <element name="TurnAround"
      type="qwsdl:TurnAroundType"
      minOccurs="1" maxOccurs="1"/>
  </sequence>
  <attribute name="Qualification"
     type="QualificationType"/>
  <attribute name="Offered" type="boolean"/>
</complexType>
<complexType name="ServiceTimeType">
  <attribute name="value" type="float"/>
  <attribute name="unit" type="string" fixed="sec"/>
  <attribute name="qualifier"
     type="qwsdl:QualifierType"/>
  <attribute name="direction"
     type="qwsdl:DirectionType" fixed="decreasing"/>
  <attribute name="source" type="qwsdl:SourceType"/>
</complexType>
<complexType name="TurnAround">
  <attribute name="value" type="float"/>
  <attribute name="unit" type="string" fixed='sec'/>
  <attribute name="qualifier"
     type="qwsdl:QualifierType"/>
  <attribute name="direction"
     type="qwsdl:DirectionType" fixed="decreasing"
  <attribute name="source" type="qwsdl:SourceType"/>
</complexType>
```

2 The datatypes TypeType, DirectionType and SourceType, not listed in the portion of Q-WSDL XML Schema, are enumerations datatypes that specify the set of acceptable values for attributes *type*, *direction* and *source*, respectively.

The Q-WSDL XML Schema also validates standard WSDL documents, because elements of QoS characteristics and associated dimensions are defined as optional elements.

An appropriate trade-off between completeness and usability has driven the derivation of the Q-WSDL metamodel in Figure 2. Additional classes, attributes and associations could be introduced to describe, e.g., more QoS characteristics or more details about the service's hosting environment or service consumer's application, although it is likely that the introduction of additional details would result in adding excessive complexity to a relatively simple but powerful language like WSDL, thus preventing an effective use of Q-WSDL.

For a similar reason, the Q-WSDL metamodel does not include elements that address advanced QoS *management* activities, such as the explicit specification of QoS contracts, QoS levels and QoS adaptation criteria. Such additional issues could easily be included into Q-WSDL by following the same process described in Section 3, or better added on top of Q-WSDL by use of additional specialized languages (e.g., WSLA) that exploit Q-WSDL without cluttering its design.

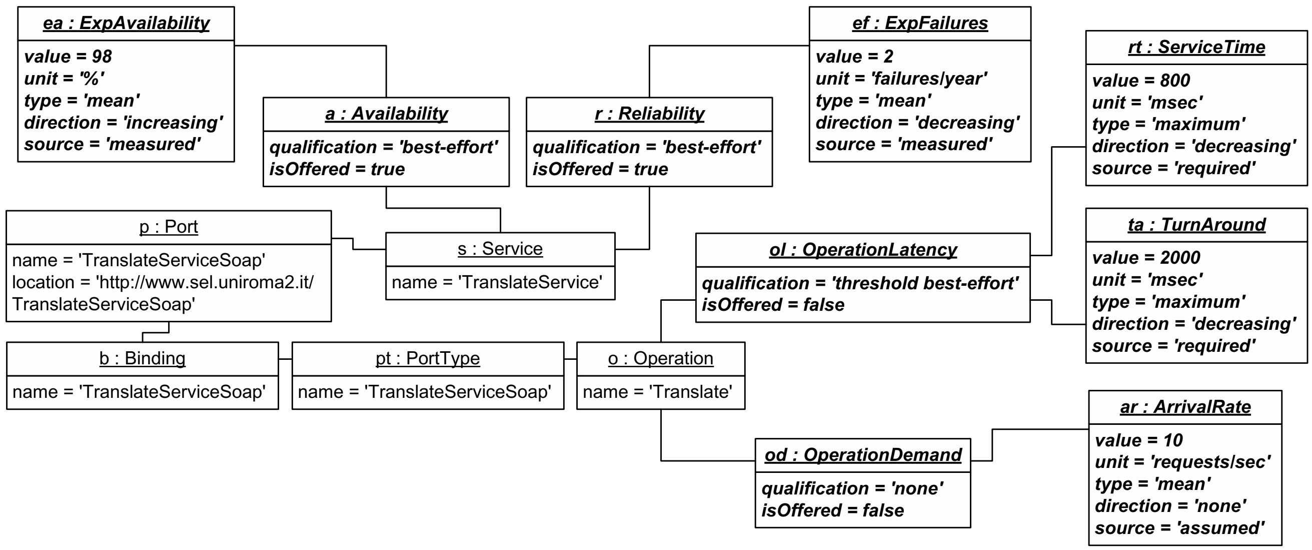

Figure 3. Q-WSDL example application

6. Q-WSDL Example Application

As said in Section 1, the proposed Q-WSDL metamodel can be applied to serve different purposes. This section gives an example application of Q-WSDL, namely to the QoS specification from the point of view of both service consumers and service providers.

Specific Q-WSDL documents are here represented as Q-WSDL models (see Figure 1), or object diagrams instantiated from the Q-WSDL metamodel in Figure 2.

Figure 3 illustrates the use of Q-WSDL for specifying offered and required QoS of an example web service that provides an operation to translate documents[3].

The example describes a usage scenario that assumes a mean arrival rate of 10 requests/sec for the `Translate` operation, as quantified by the object of `ArrivalRate` type linked to the QoS characteristic object of `OperationDemand` type.

Performance properties of the operation provided by the web service are specified as QoS characteristics required by the service consumer. In particular, the objects of `ResponseTime` and `TurnAround` type quantify the QoS characteristic object of `OperationLatency` type. The object of `ResponseTime` type specifies a constraint on the maximum time to execute the operation at server side (800 msec), while the object of `TurnAround` type specifies a requirement of maximum 2 seconds to complete the operation, including the time to transfer requests and responses from and to the service consumer (without regard to type and capacity of the network linking service consumer to service provider).

Service reliability and availability are instead specified as QoS characteristics offered by the service provider (see the value of attribute `isOffered` on objects of `Reliability` and `Availability` type), which specifies measured mean values of expected number of failures and expected availability, respectively.

The example does not specify any assumption or requirement about the network characteristics, the encryption of messages or the access control policy of the web service, and thus no objects are instantiated from classes `Network` `Encryption` and `AccessControl`, as well as from the classes linked by use of *QoS dimension* associations.

Objects in the object diagram are directly translated into XML elements, according to XMI rules for XML document production. The resulting Q-WSDL XML document can then be validated by use of the Q-WSDL XML Schema obtained from the Q-WSDL metamodel, as illustrated in Section 5.

3 For the sake of clarity, object diagrams of this section only consider WSDL objects relevant to the Q-WSDL extension and objects specific to Q-WSDL are shown in ***bold-italic*** characters.

7. Conclusions

Web services are the building blocks of the emerging computing paradigm based on service-oriented architectures. Web service capabilities are expressed in XML by use of the Web Service Definition Language (WSDL).

Unfortunately, a WSDL description only addresses the functional aspects of a web service without containing any useful description of non-functional or quality of service characteristics.

This paper has introduced a lightweight WSDL extension for the description of QoS characteristics of a web service.

The WSDL extension, called Q-WSDL (QoS-enabled WSDL), is inspired by the OMG QoS and SPT Profiles and has been carried out as a metamodel transformation, according to principles and standards provided by MDA.

An example application to the specification of QoS requirements has been illustrated, but Q-WSDL can also be used to add QoS characteristics when querying registries of web services, to define the SLS part of a SLA, to enable the QoS-aware composition of web services and, being a model-driven extension, to support the automated mapping from WSDL documents to Q-WSDL ones and from UML models to Q-WSDL web services.

References

[1] J. Bezivin, S. Hammoudi, D. Lopes, F. Jouault, *An Experiment in Mapping Web Services to Implementation Platforms*, Atlas Group, INRIA and LINA University of Nantes, Research Report, March 2004.

[2] B. Bordbar and A. Staikopoulos, Automated Generation of Metamodels for Web service Languages, *Proceedings of the Second European Workshop on Model Driven Architecture (MDA)*, Canterbury, UK, September 7–8, 2004.

[3] D. Box, F. Curbera, M. Hondo, C. Kale, D. Langworthy, A. Nadalin, N. Nagaratnam, M. Nottingham, C. von Riegen, J. Shewchuk, Web Services Policy Framework (WS-Policy), http://www.ibm.com/developerworks/library/ws-policy, May 28, 2003

[4] N. Catania, P. Kumar, B. Murray, H. Pourhedari, W. Vambenepe, K. Wurster, Web Services Management Framework, Version 2.0, Hewlett-Packard, http://devresource.hp.com/drc/specifications/wsmf/WSMF-WSM.jsp, 2003.

[5] A.D'Ambrogio, A Model Transformation Framework for the Automated Building of Performance Models from Software Models, *Proceedings of the Fifth International Workshop on Software and Performance (WOSP 2005)*, July 11-15, 2005, Palma de Mallorca, Illes Balears, Spain.

[6] D. Frankel, J. Parodi, *Using Model-Driven Architecture to Develop Web Services*, IONA Technologies PLC White Paper, 2nd Ed., April 2002.

[7] International Organization for Standardization, CD15935 Information Technology: Open Distributed Processing - Reference Model - Quality of Service, ISO document ISO/IEC JTC1/SC7N1996,(October 1998).

[8] H. Ludwig, A. Keller, A. Dan, R. King, R. Franck, Web Service Level Agreement (WSLA) Language Specification, Version 1.0, IBM Corporation, http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf, Jan. 2003.

[9] H. Ludwig, Web Services QoS: External SLAs and Internal Policies - Or: How do we deliver what we promise?, *4th IEEE International Conference on Web Information Systems Engineering*, WISE 2003 Workshops, Roma (Italy), 13 December 2003.

[10] M.R. Lyu (ed.), *Handbook of Software Reliability Engineering*, McGraw-Hill, 1995.

[11] D. Martin, M. Burstein, O. Lassila, M. Paolucci, T. Payne, S. McIlraith, *Describing Web Services using OWL-S and WSDL*, `http://www.daml.org/services/owl-s/1.1/owl-s-wsdl.htm`, 2004.

[12] D.A. Menascé, QoS Issues in Web Services, *IEEE Internet Computing*, pp. 72–75, Nov./Dec. 2002.

[13] Object Management Group, *MDA Guide*, version 1.0.1, June 2003.

[14] Object Management Group, *Meta Object Facility (MOF) Specification*, version 1.4, April 2002.

[15] Object Management Group, *Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification*, Final Adopted Specification, November 2005.

[16] Object Management Group, *UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms*, Adopted Specification, May 2005.

[17] Object Management Group, *UML Profile for Schedulability, Performance and Time Specification*, v. 1.1, January 2005.

[18] Object Management Group, *Unified Modeling Language (UML) Specification*, version 1.4, September 2001.

[19] Object Management Group, *XML Metadata Interchange (XMI) Specification*, version 2.0, May 2003.

[20] M.P. Papazoglou, D. Georgakopoulos, Service-oriented computing, *Communications of the ACM*, vol. 46, no. 10, October 2003, pp. 25-28.

[21] D.B. Petriu, M. Woodside, A Metamodel for Generating Performance Models from UML Designs, *Proceedings of UML2004*, Lecture Notes in Computer Science 3273, Lisbon, Portugal, October 11-15, 2004.

[22] M. Tian, A. Gramm, H. Ritter, J. Schiller, A Survey of current Approaches towards Specification and Management of Quality of Service for Web Services, *PIK Journal*, vol. 3, 2004, July-August 2004, pp.132-139.

[23] V. Tosic, B. Pagurek, K. Patel, WSOL A Language for the Formal Specification of Classes of Service for Web Services, *Proceedings of the 2003 International Conference on Web Services*, Las Vegas, USA, pp. 375-381, 2003.

[24] K. Verma, K.Sivashanmugam, A. Sheth, A. Patil, S. Oundhakar, J. Miller, METEOR-S WSDI: A scalable P2P infrastructure of registries for semantic publication and discovery of web services, *Inf. Tech. and Management*, 6(1):1739, 2005.

[25] WWW Consortium, Web Services Description language (WSDL) Version 2.0, W3C Working Draft, January 2006, `http://www.w3.org/TR/wsdl20/`

[26] WWW Consortium, *XML Schema*, W3C Recommendation, `http://www.w3.org/ XML/Schema`.