CONTRACTS FOR DEFINING QOS LEVELS IN A MULTICHANNEL ADAPTIVE INFORMATION SYSTEM

M. Callea, L. Campagna, M.G. Fugini, P. Plebani
Politecnico di Milano
20133 Milano Italy
moreno.callea@tin.it, luca.campagna@tiscali.it, fugini@elet.polimi.it, plebani@elet.polimi.it

Abstract
Multichannel Information Systems provide a way to invoke the same service through several channels. In this way, even if the functionality provided by the service is independent from the actual channel, the quality varies according to the particular devices used by the consumer.

In this context this discusses about the creation and management of a contract formalizing the agreement, in terms of quality of service parameters, between an e-Service provider and an e-Service consumer within a multichannel adaptive information system. In particular the work relates to some of the existing modeling languages for the QoS, such as QML, WSLA and XQoS, that are extended in order to adapt them to a multichannel environment. The presented model is validated by a prototype developed to support creation and subscription of contracts. The application allows a provider and to publish offers regarding e-Services with desired QoS parameters, and a consumer to subscribe a contract with the negotiated QoS levels.

1. Introduction
The first efforts around Service Oriented Computing (SOC) mainly focused on the definition about how an e-Service could be built, deployed, and invoked. As a consequence, different standards or standard proposals are now available, and different platforms are able to provide a set of e-Services mainly through the Web [Alonso et al., 2003].

Starting from this scenario, most recent researches concentrates on possible extensions of the Service Oriented Architecture (SOA) where the coordination, composition, and management of e-Service are also considered [ACM, 2003]. Besides such conceptual extensions, starting from the consideration that the Web Service technology is an instantia-
tion of the SOA, it is very interesting to consider other kinds of extension in order to define how an e-Service can be provided through several channels using for example a Smartphone or a PDA. In this way, if the same e-Service can be exploited through several channels, then the user can decide to switch among such available channels due to the actual needs. In particular, in this work, we associate these needs to the quality of e-Service in order to allow the user to change the e-Service exploiting when he realizes that the actual quality is considered not enough.

According to various proposals currently available to define the QoS, this paper deals with the need of models and methods that allow the specification of quality levels in Multichannel Information Systems. In particular, in the Italian MAIS\(^1\) project [The MAIS Team, 2003], multichannel systems are regarded as able to provide an e-Service on different channels having different technological characteristics, such as diverse delivery times, responses, or simply different data rendering, depending on the used protocols, networks, and devices. On the other hand, Adaptive Systems are regarded as able to analyze the network and to suggest the user the most convenient way (e.g., the most suitable receiving device, or the most suitable transmission mode) to receive and use the requested e-Service maintaining an adequate quality level.

In this paper, we present the model studied in the MAIS project for specifying the QoS in Multichannel Adaptive Information Systems. In this model, a user which is looking for an e-Service around the network is interested not only on the functional aspects, but also on the non-functional aspects of the e-Service, such as response time, security and integrity of transactions, or costs, which can be grouped under the term of Quality of Service (hereafter QoS) aspects.

The paper is organized as follows. After a brief analysis on some existing QoS modeling languages presented in Section 2, Section 3 discusses about the MAIS approach, whereas Section 4 presents an extension to the existing languages to cover aspects critical to multichannel system. This, Section 5 describes the prototype developed on the basis of the presented model and outlines about future developments concludes the work.

2. Related work

QoS is considered as an important topic in several research communities and a lot of work had been done to provide a definition. For this reason, nowadays several languages and specifications are available in telecommunication [(ITU), 1994; (ITU), 2001; Crawley et al., 1998; Huston, 2000], middleware [Zinky et al., 1997; Marchetti et al., 2003], and
information system communities [Frølun and Koistinen, 1998; Keller and Ludwig, 2002; Exposito et al., 2003]. Concentrating on a subset of such specifications, proposals like QML, WSLA, and XQoS, capture the main aspects to take into account in the definition and management of the QoS and, for this matter, we give an overview about them in the following.

2.1 QML (Quality of Service Modeling Language)

QML tries to model the QoS as independent as possible to the specific domain in which the service works. For this reason such a specification relies on the definition of QoS parameters organized according to the object-oriented paradigm concepts. Among the others, the QML specification lists a set of elements that every QoS documents should consider in order to provide a good specification about the quality. In particular:

- QoS specification should be syntactically separated from the other portions of service specification, such as interface definitions;

- it should be possible to specify both the QoS properties required by the user and the QoS properties about the service provisioning;

- there should be a way to determine how the QoS specification can match the user QoS requirements;

- it should be possible to redefine and to specialize an existing specification, analogously to the inheritance in the object-oriented programming.

According to these requirements, QML provides three main abstraction mechanisms for QoS specification: contract type, contract, and profile. While a contract type defines the dimensions that can be used to characterize a particular QoS aspect, a contract is an instance of a contract type and represents a particular QoS specification. In particular a contract type defines a collection of dimensions, each of them associated to a range of allowed values. A contract redefines these constrains according to given needs. A profile associates the contracts to the service interfaces operations, operation arguments, and operation results.

On the contrary QML, does not specify either how to enforce and to monitor the QoS or how the responsibilities are distributed among the involved actors.
2.2 WSLA (Web Service Level Agreement)

WSLA is an XML-based, extensible language used to define a contract between a Web Service provider and respective Web Service user. Analogously to QML, WSLA defines QoS levels according to a set of different quality parameters but, despite the QML also considers the responsibility about the monitoring and enforcement of the quality. Briefly, a WSLA document is composed by three main sections:

- parties description: who is involved in the contract;
- service definition: what are the parameters describing the QoS, what are the metrics related to them, and for each parameter who is in charge to monitor the values;
- obligation: the range of values the parameters have to respect and the action to be taken in case of violation.

Due to its native purpose, WSLA is strictly related to the Web Service provisioning, and it is not able to specify the QoS when the same service is provided through a channel different from the Web.

2.3 XQoS (XML-Based QoS Specification Language)

XQoS defines the QoS on both user and provider standpoint where the multimedia service is the class of service mainly considered. At the user side the parameters are bound to the human perception of a service, whereas at the provider side these parameters are bound to the communication services used to provide the service. This specification relies on a formal model represented by Time Stream Petri Nets [RIFERIMENTO] for multimedia systems.

Even if XQoS is too much related to a particular class of application, i.e. multimedia application, the provided modelization about the elements which composes a multimedia service can be reused inside a multichannel system.

3. Quality model

In the MAIS project, the problem of defining QoS is one of the main topic since, as discussed in the introduction, the quality definition enables the adaptivity definition. Figure 1 sketches the quality model adopted in MAIS [Marchetti et al., 2004] on which the contract, subject of this work, must base. In particular, this model studies how the channel can influence the quality perceived by the user with respect to the
Contracts for defining QoS levels in a Multichannel Adaptive Information System

quality provided by a system. To reach this goal, the model consists of (i) a system model, defining objects (e-Service, network and device) and actors (e-Service provider, network provider, device provider, and user), and of (ii) a set of roles and rules enabling the association between quality information, expressed by quality parameters, and objects.

In this way the ⟨network, device⟩ pair represents the channel able to connect the e-Service provider and the user. In particular, given an e-Service, the e-Service provider selecting the networks and the devices defines a set of channels through which the e-Service can be invoked. Examples of channels are < 802.11, PDA >, < Internet, PC >, < 802.11, PC >, and < GSM, Smartphone >.

In order to attach quality information to the objects, the model introduces the quality parameters as the pair < name, admissible_value > where the name represents a unique identifier of the parameters, whereas the admissible_value the range of values suitable for this particular parameter.

In order to normalize the possible different interpretations about the concepts related to quality, the model introduces the community as a group of providers proposing a specification for a group of objects with some relevant common characteristics. Hence, we have a e-Service community for groups of e-Services providing the same functionality (e.g. hotel reservation service, video on demand service), a set of network communities and device communities. In this way the communities declare both the functional specifications and set of quality parameters they consider relevant, and all the providers who aim at implementing the relative object will refer to such specifications.

To better explain the model, let us consider the example about a video-on-demand e-Service, which allows a user to receive video-streams on several heterogeneous devices, e.g., traditional PCs, cable TVs, or SmartPhones. Both the functional and quality features of such a class of e-Service are specified by the video-on-demand e-Service community. In particular, the quality of the video-on-demand e-Service can be characterized by quality parameters such as framerate, colordepth, and resolution. In the same way let use suppose that a network community defines the quality of a generic network according to the bandwidth, latency and jitter quality parameters, whereas the devices community introduces the videoresolution and colordepth parameter. For each of these parameters, the communities also define the range of admissible values.

Once a provider decides to implement and to make an object available, he has to define the quality according to the parameters specified by the related communities. This definition means a restriction of the range of allowed values of quality parameters describing the e-Service, obviously
respecting the guidelines of the e-Service communities. For example, if the video-on-demand e-Service community has defined as admissible values for the framerate, the range [5 fps..40 fps], the e-Service provider, on the basis of the available computational resources, can restrict such a range to the value he is actually able to provide for the e-Service to the range [5 fps..20 fps].

So far, the quality is defined from the provider perspective. Considering the user perspective (in this case the user of the e-Service), we are not sure that the quality defined by the e-Service provider will be the same perceived by the user [Khirman and Henriksen, 2002]. In fact, the network and the device affect in same way the e-Service exploitation. In the MAIS quality model, such an influence is captured by the quality rules that explicit the relationship between the different quality parameters, in order to compute the quality of experience, i.e., the actual quality perceived by the consumer. For example the following quality rule states the relationship among one of the network parameters, i.e., the bandwidth, and the e-Service quality parameters.

\[
\text{framerate} \ast \text{colordepth} \ast \text{resolution} \ast = K \ast \text{bandwidth}
\]

In this way, it is possible to compute the minimum and the maximum values of the framerate parameter allowed from the user side. Identifying and computing such quality rules for all the QoS parameters gives the user the set of elements to decide which is the best channel where an e-Service is executed in different situations.

\[\text{Figure 1. MAIS model}\]
4. WSLA extension

The quality model described above allows the contract definition, where a contract is a document in which two parties set up an agreement. Such a document must contain: all the data identifying the contracting parties, the object of the contract, the general conditions of agreement, and the responsibilities in case the object of contract is not fulfilled.

These aspects are well captured by the WSLA language described above, but some extensions are needed. In particular such extensions refer to the concepts belonging to QML and XQoS in order to describe the quality not only on the Web Service environment but also on the multi-channel environment.

Entering in detail, a first kind of extension introduces some attributes to the ServiceLevelObjective and ActionGuarantee tags which namely represent the admissible values of quality parameter, and the actions to be taken in case of violation.

In particular the attribute order, which can be increasing or decreasing, states how the quality varies with respect to an increasing value of the quality parameters. This specification is needed since the order depends on the semantics of parameters; for example, for the response time, the higher the value the lower the quality, whereas, for the throughput, the higher the value the higher the quality.

About the channel definition, introducing the attributes device and network, the contract can now specify a particular quality level which depends on the selected channel. In this way, it is possible not only to characterize an e-Service by the available channels but also to compare the different qualities in order to enact the more suitable adaptation strategies.

The last kind of extension is about the introduction of two new kinds of domains not present in WSLA: the set domain and the enumerated domain. In both cases a quality parameter must assume one of the values belonging in the domain but, in the former a quality parameter can assume one or more of the value at the same time, whereas in the latter case one and only one.

The next paragraphs briefly describes how such extensions are used and the main section which compose a contract.

4.1 The parties

One of the most important part of the contract is about the description of the actors, also called parties, of the agreement, i.e. provider and consumer. In addiction, a third party, called guarantee, is introduced in order to control and guarantee about the contract terms between the
provider and the consumer. In our case, the third party is the actor that
measures, or can retrieve the quality parameters values, and compares
them against the values specified in the contract. Actually, depending
on the nature of the contract and the role of the parties, the contract
can also consider the provider or the user as a guarantee.

If the guarantee during the monitoring activity measures a value out-
side of the range of admissible values, then notifies the exception to the
two involved parties as shown in Figure 2. Here each party is defined by
the its name, address, and information about actions to take in case of
violation notification.

<Parties>
  <ServiceProvider name="myProvider">
    <Contact>
      <Street>21 Rome st</Street>
      <City>Milan</City>
    </Contact>
    <Action name="notification" partyName="myProvider" xsi:type="WSDL SOAPActionDescriptionType">
      <WsdlFile>notification.wsdl</WsdlFile>
      <SOAPBindingName>VideoOnDemandBindingName</SOAPBindingName>
      <SOAPOperationName>VideoOnDemandOperationName</SOAPOperationName>
    </Action>
  </ServiceProvider>
  <ServiceConsumer name="...">
    ...
  </ServiceConsumer>
  <SupportingParty name="...">
    ...
  </SupportingParty>
</Parties>

Figure 2. Parties section

4.2 Service definition

For each e-Service a set of quality parameters is attached as shown
in Figure 3. Such parameters are defined by: a measure unit, a met-
ric, the data type, and the indication of parties that can provide, read,
and manage this data. Here, the Operation tags states how to find
and to invoke the e-Service referring to the respective WSDL specifi-
cation. The Schedule tag holds the date of validity of the contract
while SLAParameter is the object storing information about the QoS
parameters. Each SLAParameter has a Metric that can be simple, or
composite. A simple metric means that its measure is provided directly
from a measurement system; hence, in the contract, the location of this
measurement system is written. Otherwise, if the metric is composite, this section shows the way to aggregate data in order to compute it.

```xml
<ServiceDefinition>
  <Operation name="GetVideo" xsi:type="wsla:WSDLSOAPOperationDescriptionType">
    <WsdlFile>servicefile.wsdl</WsdlFile>
    <SOAPBindingName>VideoOnDemandBindingName</SOAPBindingName>
    <SOAPOperationName>GetVideo</SOAPOperationName>
    <Schedule name="MainSchedule">
      <Period>
        <Start>06/02/2004</Start>
        <End>06/02/2005</End>
      </Period>
      <Interval>
        <Second>1</Second>
      </Interval>
    </Schedule>
    <SLAParameter name="Bandwidth" type="float" unit="Kbps">
      <Metric name="Bandwidth" type="float" unit="Kbps">
        <Source>Provider</Source>
        <Pull>Provider</Pull>
        <Push>Provider</Push>
      </Metric>
      <Communication>
        <Source>Provider</Source>
        <Pull>Provider</Pull>
        <Push>Provider</Push>
      </Communication>
    </SLAParameter>
    <Metric name="Bandwidth" type="float" unit="Kbps">
      <Source>Provider</Source>
      <MeasurementDirective xsi:type="wsla:Counter">
        <MasuremementURI>http://MeasurementService.com/</MasuremementURI>
      </MeasurementDirective>
    </Metric>
  </Operation>
</ServiceDefinition>
```

Figure 3. ServiceDefinition section

4.3 Guarantee terms

This section of the contract glues the parties to a particular e-Service also considering the channels the e-Service supports. Here, it is important to note that the while the set of QoS parameters strongly depends on the provided e-Service, the admissible values perceived by the user strictly depend on the channel used by the consumer. For this matter the guarantee must monitor what the user perceive and not what the system provide, thus for each `<device, network>` pair, i.e. the channel, the WSLA portion specializes the quality parameters with: the range of allowed values, the order of the allowed values, the party which has to take over in case of contract violation, and the agreement validity time. Here, the actual values are calculated starting from what the provider
offers according to the identified quality rules for the considered channels.

In particular, Figure 4 indicates, in the ServiceLevelObjective tag, the range of allowed values of each QoS parameter, the related device, and network interface. For each e-Service level exists an Action Guarantee object and the information about how to invoke the notification actions. In the example the '*' symbol means that all the involved parties have to be informed.

```xml
<Obligation>
  <ServiceLevelObjective device="Computer" name="SLBandwidth" network="802.11b" order="Increasing">  
    <Validity>
      <StartDate>06/02/2004</StartDate>
      <EndDate>06/02/2005</EndDate>
    </Validity>
    <Expression>
      <Predicate xsi:type="Greater">
        <SLAParameter>Bandwidth</SLAParameter>
        <Value>350</Value>
      </Predicate>
    </Expression>
    <Expression>
      <Predicate xsi:type="Less">
        <SLAParameter>Bandwidth</SLAParameter>
        <Value>500</Value>
      </Predicate>
    </Expression>
    <EvaluationEvent>NewValue</EvaluationEvent>
  </ServiceLevelObjective>
  <ActionGuarantee name="GDNBandwidth">
    <Obliged>*</Obliged>
    <Expression>
      <Predicate xsi:type="Violation">
        <ServiceLevelObjective>SLBandwidth</ServiceLevelObjective>
      </Predicate>
    </Expression>
    <EvaluationEvent>NewValue</EvaluationEvent>
    <QualifiedAction>
      <Party>*</Party>
      <Action actionName="*" xsi:type="Notification">
        <NotificationType>Violation</NotificationType>
        <CausingGuarantee>SLBandwidth</CausingGuarantee>
        <Network>802.11b</Network>
        <Device>Computer</Device>
        <SLAParameter>Bandwidth</SLAParameter>
      </Action>
    </QualifiedAction>
    <ExecutionModality>Always</ExecutionModality>
  </ActionGuarantee>
</Obligation>
```

Figure 4. Obligation Section

5. A prototype application for creating contracts

According to the specification described above we built a prototype (Figure 5) which allows the contract definition and the management in a multichannel environment. Here, the provider represents who is in charge to built and offer the e-Service through a set of channels. The consumer uses such an e-Service and the selection of the channel
is driven by the selection of the device. The contracts which represent the agreement between the user and the provider, defined before the e-Service invocation occurs, are stored in the Contract Repository. The Monitoring System is responsible to measure the QoS perceived by the consumer, to extract the contracts from the Contract Repository and to compare the values written in the contracts with the measured values. In case of agreement violations, the Monitoring Service invokes the suitable notification services. These services act on behalf of the provider and consumer.

As a main scenario supported by the prototype we refers to a Video-on-demand e-Service which provide to the users a set of video clips related to soccer matches. As discussed above the quality of provided clips is affected not only by the provider but also by the selected channel. In fact, even if a provider was able to broadcast with an high resolution, a user with a Smartphone can not completely appreciate such a resolution.
since the device has a limited screen size. The same about the network that, due the latency and the bandwidth values, can even block the video broadcasting.

In order to avoid such unavailability, the MAIS project to provide a set of adaptation strategies able to allow the user to switch among the channels. Actually the channel switching can be also influenced by the user preferences who can start watching the clips on one device, the PC, and finish the vision on the SmartPhone since he has to travel.

The organization of the quality information described in the community specifications and the object implementations are shown in Figure 6. All of these documents use the XML due to its portability and ease of use.

![Figure 6. Reference architecture](image)

The System Management module has three associated repositories, where the e-Services used by the application, the offers of the providers, and the contracts subscribed by consumers are stored. The Provider’s System Management refers to the Communities Repository in order to obtain the specifications defined by the community needed to implement an object compliant to them. Otherwise the Consumer’s System Management use e-Service Repository to retrieve the offers of the providers, to compare them and, once on of them is selected as the effective provider, to define and store the contract in the Contract Repository.
6. Concluding remarks and future work

This paper described a model for creating QoS contracts in a multi-channel adaptive environment, and an application supporting the creation and management of such contracts. Our approach relies on some of the existing approach available on the literature and provide a syntactical extension to WSLA in order to capture the peculiarity of the multichannel systems.

Even if the prototype is able to manage the interaction among the provider and the user in order to define the contract, it should be extended mainly on the monitoring aspects. For this matter we are studying about how we can capture the values of the quality parameters. Nearby the monitoring, a set of e-Service able to react to the quality changes is needed. So, these e-Services could be used by the other systems composing MAIS as a trigger for the adaptation strategies.

Notes

1. MAIS-Multichannel Adaptive information Systems is a National Research Project funded by the Ministry of Research, spanning 2003-2005, and involving Universities, Research Institutions, and ICT Companies.

References


