CONTRACTS FOR DEFINING QOS LEVELS
IN A MULTICHANNEL ADAPTIVE INFORMATION SYSTEM

M. Callea, L. Campagna, M.G. Fugini and P. Plebani
Politecnico di Milano - 20133 Milano Italy

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 of the actual channel, the quality varies according to the particular devices used by the service consumer. In this context, this paper presents a frame for the creation and management of contracts formalizing the agreement, in terms of quality of service, between an *e*-Service provider and an *e*-Service consumer in a multichannel adaptive information system. In particular, the work relates to some of the existing modelling languages for QoS, such as QML, WSLA and XQoS, and presents some extensions to WSLA able to deal with a multichannel environment. The presented model is validated by a prototype developed to support creation and subscription of contracts. The prototype application allows a *provider* to publish others regarding *e*-Services with desired QoS parameters, and a *consumer* to subscribe a contract with the negotiated QoS levels.

Keywords: multichannel systems; quality of service; contract; adaptivity.

1. INTRODUCTION

The first efforts in the field of 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 concentrate on possible extensions of the Service Oriented Architecture (SOA) where the coordination, composition, and management of *e*-Services are also considered [ACM, 2003]. Besides such conceptual extensions, starting from the consideration that the Web Service technology is an instantiation of the SOA, it is very interesting to consider other kinds of extensions 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, and it is up to the user to switch among available channels according to his current needs. In particular, in this work, we focus one expressing these multichannel requirements in terms of the quality of the delivered *e*-Service, in order to allow the user to change the *e*-Service delivery mode when he realizes that the current quality level is not adequate.

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 project [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 e-Service, while 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 looking for an *e*-Service around the network is interested not only in functional aspects, but also in 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 modelling languages presented in Section 2, Section 3 describes the proposed Quality model, whereas Section 4 presents an extension to WSLA to cover aspects of multichannel systems. Section 5 outiline the basic features of the prototype developed on the basis of the presented model and finally Section 6 draws conclusions.

2. RELATED WORK

QoS is currently 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 [Frølung et al, 1998; Keller et al., 2002; Exposito et al., 2003]. Concentrating on a subset of such specifications, proposals like QML, WSLA, and XQoS, capture the main aspects to be taken into account in the definition and management of the QoS and, to this aim, we present an overview of these three languages in the following subsections.

2.1 QML (Quality of Service Modelling Language)

QML [Frølung et al, 1998] tries to model the QoS as independent as possible of the specific domain where the service operates. For this reason, QML relies on the definition of QoS parameters organized according to the concepts of the object-oriented paradigm. QML specification lists a set of elements that each QoS document 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 what inheritance mechanisms do in 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 associated with a range of allowed values. A contract redefines these constraints according to given needs. A profile associates the contracts to the service interfaces operations, operation arguments, and operation results.

QML does not specify either how QoS can be enforced and monitored nor the way to distribute responsibilities among the involved actors.

2.2 WSLA (Web Service Level Agreement)

WSLA [Keller et al., 2002] is an XML-based, extensible language used to define a contract between a Web Service provider and a Web Service user. Analogously to QML, WSLA defines QoS levels according to a set of different quality parameters; differently from QML, WSLA considers also the responsibility about quality monitoring and enforcement. A WSLA document is composed of 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 of monitoring the values;
- *obligation:* the range of values the parameters have to respect, and the action to be undertaken in case of violation.

Due to its native purpose, WSLA is strictly related to Web Service provisioning and has no mechanisms to specify the QoS in case the same service is provided through a channel different from the Web.

2.3 XQoS (XML-Based QoS Specification Language)

XQoS [Exposito et al., 2003] defines the QoS on both the user's and provider's standpoints; moreover, the language is basically oriented to multimedia services. At the user's side, the parameters are bound to the human perception of a service, whereas at the provider's 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 [Diaz et al. 1994] for multimedia systems.

Even if XQoS is strictly related to a particular class of applications, i.e. multimedia, the provided modelling concepts about elements composing a multimedia service are useful for multichannel systems.

3. QUALITY MODEL

In the MAIS project, the problem of defining QoS is one of the main topics since the definition of what quality means and how it can be measured and monitored during service provisioning enables system designers and providers/users to properly define the concept of "adaptivity". Figure 1 sketches the quality model adopted in MAIS [Marchetti et al., 2004] on which the contract needs to be based. In particular, this model represents how the channel can influence the quality, as perceived by the user, with respect to the quality, as provided by the system. Hence, 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.

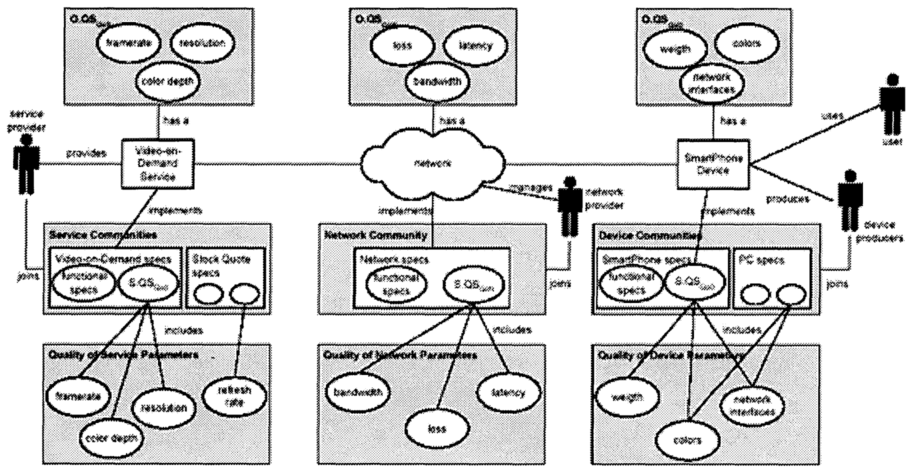

Figure 1. MAIS Model

The <*network, device*> pair represents the *channel* able to connect the *e*-Service provider and user. In particular, given an *e*-Service, the *e*-Service provider, by selecting the networks and the devices, defines a set of channels through which the *e*-Service can be invoked. Simple examples of channels are: < 802.11; PDA >,<Modem; PC >,<802.11; PC >,<GSM; Smartphone>.

In order to attach quality information to the objects, the model introduces *quality parameters* in the form of <*name; admissible value*> pairs, where *name* represents a unique parameter identifier, and *admissible value* represents the range of values suitable for the parameter. To normalize the possible different interpretations of concepts related to quality, the model introduces the *community* as a group of providers who propose a specification for a group of objects with relevant common characteristics. Hence, we have an *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*. The communities declare both the *functional specifications* and *the set of quality parameters* they consider as relevant. All the providers who intend to implement the relative object will refer to such specifications.

To clarify the model, consider an example regarding a video-on-demand *e*-Service, allowing a user to receive video-streams on different devices, e.g., PCs, cable TVs, or SmartPhones. Both the functional and quality features of this class of *e*-Service are specified by the *user community* of the video-on-demand *e*-Service. In particular, the quality of this *e*-Service can be characterized by quality parameters such as *framerate*, *colordepth*, and *resolution*. Analogously, 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* parameters. For each parameter, the communities also define the range of admissible values. Once a provider decides to implement an object, he has to define the quality according to the parameters specified by the communities related to the object. This means a possible 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 the range [*5fps*...*40fps*] as admissible values for *framerate*, the *e*-Service provider, on the basis of his available computational resources, can restrict such range to, say, [*5fps*...*20fps*], that is, to the value he is actually able to provide for the *e*-Service.

So far, the quality has been defined from the provider's perspective. Considering the *e*-Service user's perspective, we observe that, in general, the quality perceived by the user is possibly different from the quality perceived by the provider [Khirman and Henriksen 2002]. In fact, the network and the device both affect the *e*-Service exploitation. In the MAIS quality model, this influence is captured by a set of *quality rules* that make 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*.

The quality rule *framerate*colordepth*resolution=K*bandwidth*, for example, states the relationship among one of the network parameters, i.e., the bandwidth, and the *e*-Service quality parameters. In this way, it is possible to compute the minimum and maximum values for *framerate*, allowed from the user side. By identifying and executing the quality rules for all the QoS parameters, the user gets the basis for deciding the most suitable execution channel.

4. WSLA EXTENSION

The quality model described above enables providers and user to set up a *contract,* intended as a formal document where two parties set up an agreement, in our case about provisioning and usage of an *e*-Service. As a basis, such document should contain: i) the data identifying the contracting parties, ii) the object of the contract, iii) the general conditions of agreement, and iv) the responsibilities and penalties in case of violation. These aspects are properly captured by the WSLA language described above. However, some extensions are needed, in particular with concepts belonging to QML and XQoS suitable to describe the quality in a multichannel environment, rather than in Web Service environments only.

The first kind of extension we propose for WSLA introduces some attributes to the *ServiceLevelObjective* and *ActionGuarantee* tags which represent the admissible values of quality parameters, and the actions to be taken in case of agreement violation. In particular, the *order* attribute, 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.

As for channel definition, a second extension we propose for WSLA is the introduction of the *device* and *network* attributes; now, the contract is suitable to specify a particular quality level, which depends on the selected channel. This extension supports the not only the description of an *e*-Service through the available channels but also the comparison of different quality values, in order to enact the more suitable adaptation strategies for *e*-Service provisioning.

The last kind of extension regards the introduction of two new kinds of *domains* in WSLA: the *set domain* and the *enumerated domain*. In both cases, a quality parameter must hold one of the values belonging to the domain; however, for the first domain, a quality parameter can assume one or more values at the same time, whereas for the second domain, the parameter can hold one value only.

Now, in the next paragraphs, we are ready to describe how the proposed extensions are used, and then how the main sections composing a contract are structured in WSLA.

4.1 The Parties

One of the most important parts of the contract is the description of the actors, called *parties*, of the agreement, i.e., the *provider* and the *consumer*. In addiction, a third party, called *guarantee*, is introduced to control and guarantee the contract terms between the two parties. In our case, the third party is the actor that measures, or is enabled to retrieve from a log file, the quality parameters values, and compares them against the values specified in the contract. Indeed, depending on the nature of the contract and on 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 outside of the range of admissible values, he notifies the *exception* to the two involved parties, as shown in Figure 2: here each party is defined by its name, address, and by information about actions to be possibly taken 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="WSDLSOAPActionDescriptionType">
    <WsdlFile>notification.wsdl</WsdlFile>
    <SOAPBindingName>VideoOnDemandBindingName</SOAPBindingName>
    <SOAPOperationName>VideoOnDemandOperationName</SOAPOperationName>
    </Action>
  </ServiceProvider>
  <ServiceConsumer name="...">
    ...
  </ServiceConsumer>
  <SupportingParty name="...">
    ...
  </SupportingParty>
</Parties>
```

Figure 2. Parties section of the contract

4.2 Service definition

For each *e*-Service, a set of quality parameters is attached to its definition as shown in Figure 3. Such parameters are defined by: a measure unit, a metric, the data type, and the indication of parties that can provide, read, and manage these data. Here, the Operation tag states how to find and to invoke the *e*-Servicee referring to a WSDL specification. 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. For a simple metric, its measure is provided directly by a measurement system; hence, in the contract, the location of this measurement system is written. For a composite metric, this section shows how data can be aggregated in order to compute the metric.

```
<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>Bandwidth</Metric>
        <Communication>
            <Source>Provider</Source>
            <Pull>Provider</Pull>
            <Push>Provider</Push>
        </Communication>
    </SLAParameter>
    <Metric name="Bandwidth" type="float" unit="Kbps">
        <Source>Provider</Source>
        <MeasurementDirective resultType="float" xsi:type="wsla:Counter">
            <MasurememementURI>http://MeasurementService.com/</MasurememementURI>
        </MeasurementDirective>
    </Metric>
    <SLAParameter name="...">
        ...
    </SLAParameter>
    <Metric name="...">
        ...
    </Metric>
    ...
  </Operation>
</ServiceDefinition>
```

Figure 3. Service Definition section of the contract

4.3 Guarantee terms

This section of the contract glues the parties to a particular *e*-Service also considering the possible delivering channels of an *e*-Service . Here, it is important to notice that 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*. To fulfil this mismatching visions, the *guarantee party* must monitor what the user perceives rather than what the system provides; hence, for each <*device, network*> pair, i.e. for each channel, a specific WSLA portion specializes the quality parameters by defining: a) the range of allowed values, b) the order of the allowed values, c) the party which has to take over in case of contract violation, and d) the agreement validity time.

Here, the actual values are computed 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 the network interface. For each *e*-Service level, an *ActionGuarantee* object and information about how notification actions can be invoked are included. In the example, the '*' symbol means that all the involved parties have to be informed of a possible violation.

5. A PROTOYPE APPLICATION FOR CREATING CONTRACTS

According to the specification described above, we built a prototype (Figure 5) for contract definition and management in a multichannel environment. The provider represents the actor in charge of formulating and offering the *e*-Service through a set of channels. The consumer uses the *e*-Service; the selection of the channel is driven by the selection of the device.

The contracts representing the agreement between the user and the provider, and defined before an actual *e*-Service invocation, are stored in the *Contract Repository*. The *Monitoring System* is responsible for measuring the QoS perceived by the consumer, for extracting contracts from the Contract Repository, and for comparing the values written in the contracts with the measured values. In case of agreement violation, the Monitoring System invokes a set of suitable notification services acting on behalf of the provider and consumer.

As a sample scenario for the prototype, we refer to a Video-on-demand *e*-Service which provides 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, although a provider is able to broadcast images and clips with a high resolution, a user with a SmartPhone will not be able to fully appreciate the high resolution, since the device has a limited screen size. The same occurs for the network that, due latency and the bandwidth values, can even block the video broadcasting.

```
<Obligation>
  <ServiceLevelObjective device="Computer" name="SLBandwidth" network="802.11b"
  order="Increasing">
    <Obliged>Provider</Obliged>
    <Validity>
      <StartDate>06/02/2004</StartDate> <EndDate>06/02/2005</EndDate>
    </Validity>
    <Expression>
      <Predicate xsi:type="Greater">
```

```xml
            <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>
<ServiceLevelObjective name="..." >
    ...
</ServiceLevelObjective>
<ActionGuarantee name="..." >
    ...
</ActionGuarantee>
...
</Obligation>
```

Figure 4. Obligation section of the contract

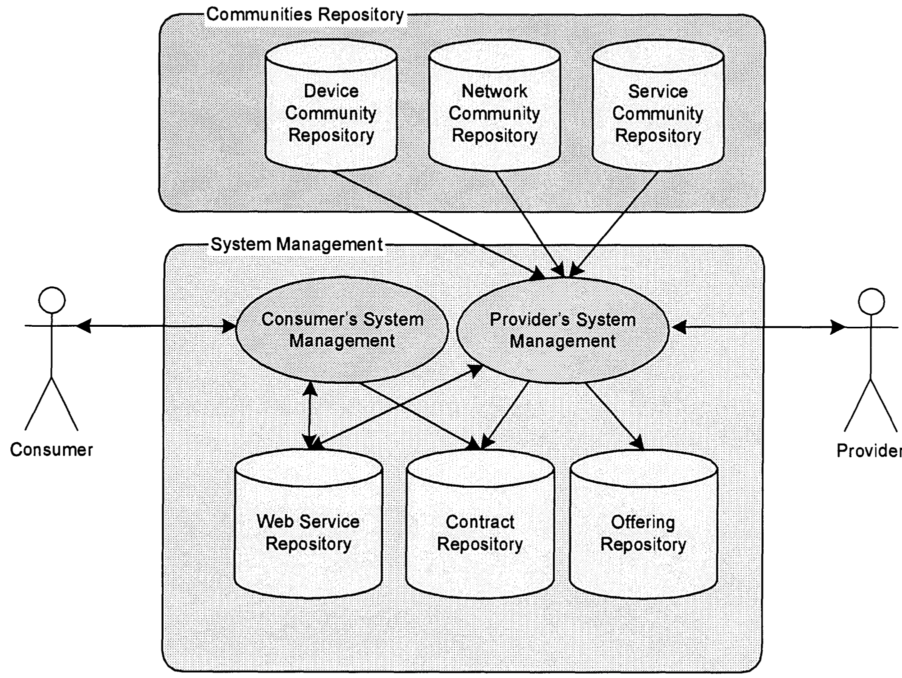

Figure 5. General Prototype Architecture for Contract Definition and Management

In order to avoid unavailability, the MAIS project is studying methods to provide a set of adaptation strategies able to allow the user to switch among different channels. Actually, channel switching can also be influenced by the user preferences. For example, the user can start watching the clips on one device, the PC, and complete the vision on the SmartPhone, e.g., because he is on travel.

The overall structure supporting quality information described in the community specifications, and the object implementations are shown in Figure 6. All of the documents are specified in XML, basically due to portability and ease of use reasons. 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.

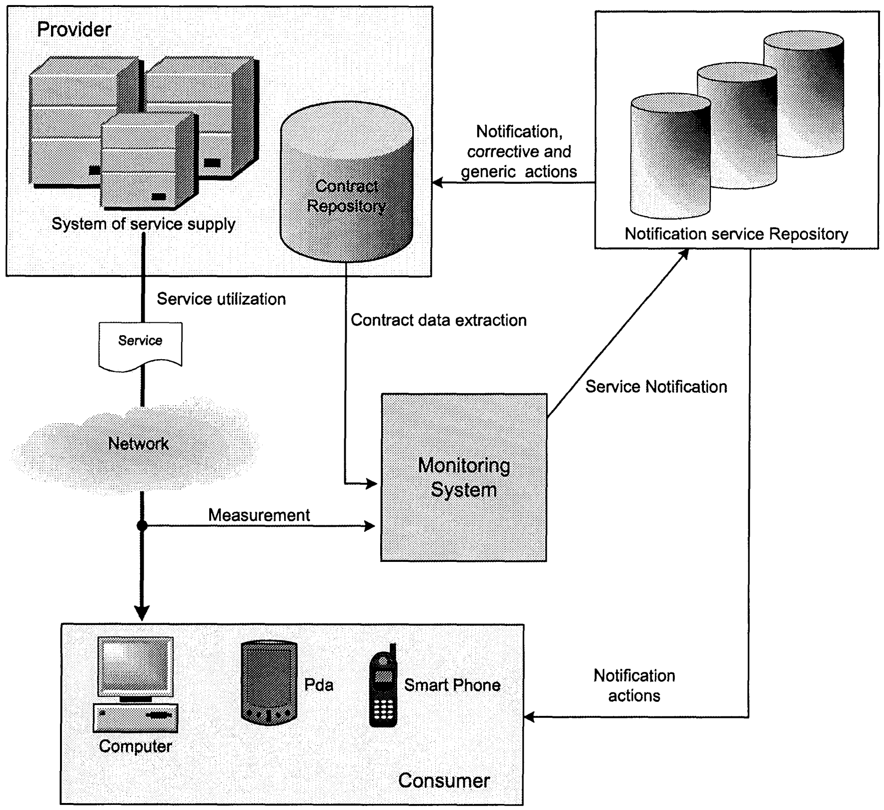

Figure 6. Reference Architecture for Quality in MAIS

6. CONCLUDING REMARKS AND FUTURE WORK

This paper has presented a model for creating QoS contracts in a multichannel adaptive environment, and an application supporting the creation and management of such contracts. Our approach relies on some of the existing approach available in the literature, and provides a syntactic extension to WSLA to capture some basic peculiarities of a multichannel system. The paper has presented some extensions to WSLA, a quality model, and a prototype supporting quality contracts creation and management.

Currently, the prototype is able to manage the interaction between provider and user in the contract definition phase; the main prototype extension regards the *monitoring aspects*. To this aim, we are investigating on the way the quality values can be captured, checked, and reasoned upon.

Besides the monitoring functions, a set of *e*-Service able to run-time react to the quality changes is needed. Moreover, these *e*-Services can be used by the other systems composing the MAIS platform as a trigger for the adaptation strategies needed by the overall MAIS architecture.

REFERENCES

ACM 2003, Communications of the ACM, Special Issue: Service-Oriented Computing, Vol. 46, n.10, October 2003.

Alonso 2003, Alonso, G., Casati, F., Kuno, H., Machiraju, V., Web Services: Concepts,Architectures and Applications (Data-centric Systems & Applications). Springer-Verlag, 2003.

Crawley et al., 1998, Crawley, E., Nair, R., Rajagopalan, B., and Sandick, H., A framework forQoS-based Routing in the Internet. Internet Engineering Task Force - RFC 2386, 1998.

Diaz et al. 1994, Diaz, M., Súnac, P., Time Stream Petri Nets - A Model for Timed Multimedia Information, in Proceedings of 15th International Conference of Application and Theory of Petri Nets 1994 Zaragoza, Spain, LNCS v.815, Springer-Verlag, 1994.

Exposito et al.2003, Exposito, E., Gineste, M., Peyrichou, R., Snac, P., Diaz, M., XQoS: XML-BasedQoS Specification Language. Proc. 9th International Conference on Multi-Media Modeling, Taiwan, January 2003.

Frølung et al. 1998, Frølung S., Koistinen, J., QML: A Language for Quality of Service Specification, Tech. Rep. HPL98-10, HP Labs, HP Software Technologies Laboratory, 1998.

Huston, 2000, Huston, G. Next Steps for the IP QoS Architecture. Internet Engineering Task Force RFC 2990. 2000.

ITU 1994, International Telecommunication Union (ITU), Terms and definitions related to quality of service and network performance including dependability. ITU Recommendation E.800 (08/94), 1994.

ITU 2001, International Telecommunication Union (ITU), Communications Quality of Service: A framework and definitions. ITU Recommendation G.1000 (11/01), 2001.

Keller et al. 2002, Keller A., Ludwig, H. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. Technical Report RC22456(W0205-171), IBM Research Division, T.J. Watson Research Center, 2002.

Khirman and Henriksen 2002, Khirman S., and Henriksen, P., Relationship Between Quality-of-Service and Quality-of-Experience for Public Internet Service, In Proceedings. of the 3rd Workshop on Passive and Active Measurement, Fort Collins, Colorado, USA, March, 2002.

MAIS Team 2003, The MAIS Team, MAIS: Multichannel Adaptive Information Systems, http://black.elet.polimi.it/mais/documenti/pdf/mais.pdf, 2003.

Marchetti et al. 2004, Marchetti C., Pernici B., Plebani P., A Quality Model for Multichannel Adaptive Information Systems, Proc. 13rd International World Wide Web WWW Conference (WWW 2004), New York City, NY, May 2004.

Zinky et al., 1997, Zinky, J.A., Bakken, D. E., and Schantz, R.E., Architectural support for quality of service for CORBA objects, Theory and Practice of Object Systems, 1997, vol. 3(1).