A quality model for e-Service based multi-channel adaptive information systems

Carlo Marchetti
Università di Roma “La Sapienza” – Dipartimento di Informatica e Sistemistica
via Salaria 113, I-00198 Roma, Italy
carlo.marchetti@dis.uniroma1.it

Barbara Pernici, Pierluigi Plebani
Politecnico di Milano – Dipartimento di Elettronica e Informazione
piazza L. da Vinci 32, I-20133 Milano, Italy
[pernici,plebani]@elet.polimi.it

Abstract

In a multichannel information system users can access the same service through different channels. At a high level of abstraction, “channel” examples are: a simple PC connected to the service provider through the Internet, a PDA connected through a wireless LAN, a SmartPhone exploiting UMTS and a private backbone. Within an adaptive multichannel information system, the system is also capable of tuning itself according to the user needs, underlying network and computing infrastructure.

We present a novel quality model for multi-channel e-Services and we propose adaptation strategies that allow describing and enforcing QoS aspects, respectively. In particular the quality model, which can be viewed as an extension of existing proposals for single-channel system, allows classifying resources of a multichannel information systems, their properties related to non-functional quality related aspects, and defines the role of the service provider as a proposer of service levels, and of the user as a selector of the adequate service levels.

1 Introduction

e-Services are self-describing components that are open and enable fast development and deployment of distributed applications. Services are exported by service providers, i.e. organizations that offer (i) the service implementations, (ii) service descriptions, and (iii) the technical and business support deemed necessary to provide the service. e-Services implementations and descriptions lays within the provider information system. Current implementation of e-Services are mainly based on Web-service, i.e. e-Services communicating through the network layers and protocols common to the Web community (TCP/IP, HTTP etc).

In a multichannel information system users can access the same service through different channels. At a high level of abstraction, “channel” examples are: a simple PC connected to the service provider through the Internet, a PDA connected through a wireless LAN, a SmartPhone exploiting UMTS and a private backbone. Depending on the channel used to access a given service, the latter can be exported with different quality levels, i.e., a channel can be associated to some level of Quality of Service (by now QoS) perceived by users exploiting it. As an example, a channel consisting of a WebTV connected by a private Intranet to a provider allows watching a video stream at a quality higher than the one achievable by a PDA connected through a wireless network. This kind of service has been recently launched on the European marketplace from relevant cellular phones network operators in preparation to the advent of UMTS and are being frequently advertised, e.g. in the soccer game domain.

Within an adaptive multichannel information system, the system is also capable of tuning itself according to the (i) user needs, (ii) underlying network and computing infrastructure. Concerning adaptation with respect to user needs, adaptation should be intended as the capability of reacting to asynchronous changes of user preferences, e.g. with respect to the channel used to access a given service. As example, a user could start watching a movie using a Smart-Phone while travelling on a train, and then, once at home, he could continue watching the movie on his WebTV. In this case the system should adapt itself to continue the service provisioning despite the user-triggered channel switch. The service level agreement (SLA) mechanisms should allow to

*This work is partially supported by the Italian MAIS (Multichannel Adaptive Information Systems) project. http://black.elet.polimi.it/mais
deal with such a class of scenarios.
For what it concerns adaptation with respect to the network and computing infrastructure, the system should be able to handle events that could determine a change in the QoS perceived by users, by reacting to such events striving to enforce the maximum QoS and to avoid violations of SLAs.
As an example, in the video-on-demand scenario the system could monitor channel behavior and then discover that the jitter of a channel exceeds a given threshold, necessary to fulfill the SLA holding with some users accessing from that channel. It could then react to such an event by increasing the compression of a stream or invite the user to switch onto another channel. Similar examples can be made considering on-line banking: a user equipped with a cellular phone with limited processing capabilities could impose the choice of a cryptography algorithm less expensive than the one implementable using a full-fledged PC.
This work aims at presenting a quality model suitable for adaptive multi-channel information systems supporting the scenarios presented above. In Section 2, after a brief introduction of the MAIS project, we introduce the e-Service and channel specifications with respect to the provider, the user and the desired adaptivity. In Section 3, a discussion about related work and research needs motivated by the multi-channel environment is presented. Section 4 and Section 5 sketch our preliminary results about our quality model and the adaptation strategies.

2 MAIS Overview

The Italian MAIS (Multi-channel Adaptive Information System) project aims at studying a set of models and methodologies which allows service provisioning through different channels.
In MAIS, a user is characterized by a context and by a profile [6] (Figure 1). The context describes, among the others, the set possible available channels and the channel currently in use, in a given location and at a given time. The profile captures user preferences which depends on a role held by the user, its expertise on the service, and a set of generic preferences that enable further service-specific user profiling. Referring to the video-on-demand example, a user could be able to watch a video on both a PC and a Smartphone, which is described in the context. Furthermore, the user could classify channels according to his preferences by stating, for instance, that when all the channels are available he prefers to use the PC when he stays at home (using the profile).
The provider is described in terms of the provided e-Services and available channels as shown in Figure 2. According to this description, an e-Service is defined as an abstraction of a functionality, or a set of functionalities, exported by a system through a standard interface. Unlike the Web Service, with the e-Service we suppose that the functionalities it performs could be invoked by different channels not only Web based. However, even if we consider the generic e-Services, for their specification we reuse the vast amount of work done in the Web-Service community.
Therefore, WSDL [10] can be used to specify the application logic, i.e. what are the operations the e-Service is able to perform, and in which way the user can invoke them. In the video-on-demand example could be operation as select video, start video, and pause video.
Proposal such as BPEL4WS [12] or WSCL [5] allow defining, at different detail levels, the e-Service behavior enabling the Web Service composition. In particular we are interested on the observable behavior [3] (named also conversation), i.e. the order in which the e-Service operations
can be invoked. DAML-S [11] can also be used to specify behavior in terms of pre and post-conditions. In the video-on-demand example, the behavior description can state that a video can be played only after it has been selected.

Using a simplified version of [22], a channel is defined as a triple \langle device, network, application protocol \rangle (see Figure 1), where device characterizes the user’s equipment (e.g. PDA, PC, Smartphone, ...), network identifies the data-link protocol exploited by the user’s device to connect to the provider (e.g. UMTS, GPRS, ...), and application protocol represents the protocol used for message exchange (e.g. HTTP, SMTP, SOAP, ...). Self-explaining examples of channels are: \langle PC, Ethernet, HTTP \rangle, \langle PC, ISDN-TA, HTTP \rangle, \langle Smartphone, UMTS, HTTP \rangle, \langle PC, 802.11b, FTP \rangle. Using channels, the service provider characterizes services (Figure 2) with the list of channels users can exploit to access the service. As different channels usually perform differently, service providers can offer services at different quality levels (the so-called service levels). Actually, service levels depend on both the service quality parameters (e.g. video frame rate, video resolution) and channel quality parameters (e.g. bandwidth, latency).

Given a description of available resources, the provider is able to quantify service levels and to export related information to users through a Service Level Specification (SLS). Then, when a user wishes to access a service, a Service Level Agreement (SLA) describing the QoS obligations of the provider (and the possible penalties) is negotiated among the user and the provider. Once a user and a provider agree on an SLA, the service provisioning phase starts. During this phase SLA must be honored as long as (i) the necessary resources stays available and (ii) the user continues to access the service from the initial channel. To attain this, the service provider system strives to monitor and control the usage of its internal resources in order to maintain the service levels agreed with users. However, in the general case channels are out of the control of the service provider, and thus their service levels could abruptly reduce hindering the maintenance of a set of SLAs. Furthermore, users can decide to switch onto different channels due, for example, to a change of the device used to access a service or network failure. Therefore, we want our system to dynamically adapt SLA upon upon facing these events. The main issue to be addressed in this context is how to continue providing the service at a service quality which is adequate to the user needs.

Addressing this issue requires to tackle a number of problems, e.g.:

- to define how service levels can be expressed;
- to define how to monitor and control resources, especially channels, and in particular which attributes can be observed and which can be controlled;
- to define means to translate requirements expressed at high abstraction level (e.g., frame rate) more useful for the user, into requirements on resource usage (e.g. bandwidth, latency) and vice-versa.

To solve the above mentioned problems, we start from the definition of a quality model for e-Services in a multi-channel information system. This model enables to reason about both QoS as to be expressed in Service Level Specifications (high level) and QoS of resources needed to enforce service levels (low level), especially of channels, that are explicitly represented in the model. Furthermore, the model makes it possible to reason about attributes of services and resource, and to devise how mapping among them can be addressed. Finally, we devise means to feed the low level of the model in order to define and implement adaptation strategies, that enable the system to react to asynchronous events as user-driven channel switches as well as to resource shortages.

3 Related work

Definition of QoS in a web-service based system is a topic largely discussed in the literature. An interesting framework which considers the main aspects interesting the QoS definition and management for the Web-Service is provided in [19]. In this proposal, the QoS is defined at two different levels (Figure 3).

At the bottom level the provider, using specific metrics, can measure the QoS of the resources exploited to provide the service and can define such QoS trough a set of quality parameters.

At the higher level, through the definition of a set of Service Levels, the provider advertises which is the possible QoS configuration he is able to guarantee for the e-Service he provides. In such a way, the user analyzes and selects a set of Service Levels creating the Service Level Agreement (SLA).

During provisioning, the satisfaction of SLA by the provider is monitored by the user or by a trusted third-party. The provider performs a set of strategies in order to enforce the promised QoS.

About the QoS definition for the e-Service, current work take into account only the service provisioning through the Web. [21] identifies the QoS parameters which are considered useful for the service provider to characterized its own Web-Service with respect to the user. [28] proposes a set of metrics which allow not only measuring the QoS parameters for a single e-Service, but also calculating measures for derived parameters for a composition of e-Services. About the quality of the data exported by the e-Service, [9] lists the main parameters (e.g., accuracy, completeness, consistency, reliability) and the related metrics. At higher level, [24] and
[16] identify how to define an SLA and which are the most important parameters which have to be considered. According to this framework, QoS definition and monitoring at resource level and service level are required.

For what it concerns QoS management with respect to the resources, both the middleware and agent communities offer significant contributions enabling the enforcement of several distinct QoS aspects. As examples, in the middleware for distributed objects community, the Common Object Request Broker Architecture (CORBA, [14]) specification details standard interfaces enabling the implementation of applications with QoS guarantees. Examples are: Real-Time CORBA (implemented by the TAO object request broker [26]) provides standard interfaces to enforce timeliness properties, e.g. scheduling policies and bounded message transfer delays; Fault-Tolerant CORBA (implemented in the Eternal [23] and IRL [4] systems) allows the enforcement of availability properties, e.g. increasing the mean time to failure, to repair and thus between failures of CORBA objects through software replication techniques; CORBA Security [15] allows the enforcement security properties such as authentication and confidentiality of object requests. Other relevant proposals, e.g., BBN’s Quality Objects, [29], entail timeliness, availability, and security properties by providing a uniform (unfortunately non-standard) framework simplifying design of mission critical applications. In the agent community, significant standardization efforts are those from the Foundation for Intelligent Physical Agents [27] (FIPA), guiding implementation efforts, e.g. JADE [7] and FIPA-OS [2]. In particular [1] defines a Monitor Agent and a Control Agent that implements QoS properties related to performance and security exploiting resource adaptation, which is achieved by observing and controlling system behavior. Finally, commercial solutions are currently targeted to enterprise QoS and system management architectures for web-service provisioning by focusing on load balancing, access control and network management, e.g. [25, 18, 17, 20].

Even if several of the above mentioned contributions deals with communication channels by taking into account common QoS properties (latency, bandwidth, jitter, cryptography etc), none of them is based on a framework for multi-channel information systems. As a consequence, objects/agents/web-services are usually not designed taking into account users that possibly access a service from several channels, and thus QoS enforcement strategies usually assume users to statically connect to a provider via a single channel, which is the same for all users.

Starting from the existing proposals, our model aims at considering a more extended environment in which the channel influences the QoS definition and management.

### 4 Quality model

Existing proposals highlight the different role held by the user and the service provider in the QoS definition. Taking into account such distinction, our quality model is defined by the Service Level Specification (SLS) as the definition of QoS offered by the service provider and the Service Level Agreement (SLA) as the agreement between the user and the provider.

From the service provider perspective, it is possible to define the QoS of his services by analyzing the resources that he allocates in order to provide the service. In our model, represented in Figure 4, we consider (i) physical resources, and (ii) data resources. With the physical resources, we consider the quality aspects of devices like CPU, disks, and memories which are used to perform the service functionalities. For data resources, we are interested in quality dimensions like currency, timeliness, and availability of data as expressed in [9]. Based on some of the resource quality dimensions, the service provider is able to identify the parameters which define the QoS strictly related to the service functionalities.

Nearby such resource related dimensions, even the QoS related to service provider and the available channels are used to define the overall QoS. About the provider, typical dimensions are related to the economical issues defining, for instance, the pricing policy for the service and the payment forms. On the contrary, the channel capture dimensions like bandwidth, latency, and jitter.

It is important to note that several quality dimensions, in particular the channel quality parameters, represent elements the provider can only monitor and cannot control and especially can be “guaranteed” only by best-effort strategies.

For this matter, we make a distinction between the QoS parameters which are both observable and controllable the

![Figure 3. General framework](image-url)
QoS parameters which are only observable. In fact, the first kind of parameters, can be used to effectively propose an SLS in order to expose quality dimensions that the provider is able to really guarantee modifying their values. On the contrary, the second kind of parameters is used only to inform the user about the possibility of monitoring a particular dimension.

On the basis of the quality dimensions introduced, the provider builds the Service Level Specification (SLS) as represented in Figure 4, expressed using high-level, user-oriented values.

Since the QoS depends on the channel, we can associate a representation of the QoS to the WSDL specification of the e-Service, according to the distinction made in [8]. In this way two WSDL specifications exist: (i) the WSDL interface document, in which only the types, messages and portType are expressed, and (ii) the WSDL implementation document in which the portType is specialized with respect to a particular binding (channel) and in which our SLS representation can be associated.

SLS can be expressed using several languages. An XML-based example for the video-on-demand e-Service is shown in Figure 5, where using the RTSP (a video streaming protocol) two different channels are available and, for each of them three different service levels (high, medium, low) are introduced. In particular, service level depends on parameters related to the device (framerate and resolution) and the provider (price and availability).

Once the SLS is defined, the user is able to decide which is the better e-Service with respect to its quality. Before the actual e-Service invocation, the user can negotiate with the service provider in order to define in which way the service will be used. The result of such phase is a contract specified by the Service Level Agreement (SLA).

According to [19], the SLA has to take into account several aspects related to the service provisioning in order to describe all the situations in which the provider and the user can be. About the definition of the value of QoS parameters, it is important to notice that the SLS could be used only as

```xml
<levels channel="PC-Eth-RSTP">
  <level id="high">
    <framerate unit="fps">
      <max>40</max><min>31</min>
    </framerate>
    <minresolution>1024x768</minresolution>
    <price currency="EUR">100</price>
    <availability unit="\%">95</availability>
  </level>
  <level id="medium">
    <framerate unit="fps">
      <max>30</max><min>11</min>
    </framerate>
    <minresolution>800x600</minresolution>
    <price currency="EUR">90</price>
    <availability unit="\%">95</availability>
  </level>
  <level id="low">
    <framerate unit="fps">
      <max>10</max><min>5</min>
    </framerate>
    <minresolution>320x200</minresolution>
    <price currency="EUR">80</price>
    <availability unit="\%">95</availability>
  </level>
</levels>
<levels channel="SPhone-UMTS-RSTP">
  <level id="high">
    <framerate unit="fps">
      <max>30</max><min>21</min>
    </framerate>
    <minresolution>320x200</minresolution>
    <price currency="EUR">200</price>
    <availability unit="\%">95</availability>
  </level>
  <level id="medium">
    <framerate unit="fps">
      <max>20</max><min>16</min>
    </framerate>
    <minresolution>320x200</minresolution>
    <price currency="EUR">180</price>
    <availability unit="\%">95</availability>
  </level>
  <level id="low">
    <framerate unit="fps">
      <max>15</max><min>5</min>
    </framerate>
    <minresolution>320x200</minresolution>
    <price currency="EUR">160</price>
    <availability unit="\%">95</availability>
  </level>
</levels>

Figure 5. A Service Level Specification for the Video-on-Demand example.
a reference point by the provider and the user during the negotiation. In fact the real value of the QoS parameter that the provider promises to the user can be different from the value presented in the SLS due, for instance, to the different bargaining power of the potential users. For this matter, before issuing an SLA between a user and a provider, an SLS can be dynamically adapted to user preferences and system resource, thus building an SLS that takes into account the user’s context and profile, as well as the available resources. As a consequence, the SLA will probably be satisfactory for users and implementable by the system.

An important issue not considered in this paper and that will be developed in future work, is the definition of penalty clauses, i.e. the behavior of the user and the provider in case of contract unfulfillment. About this issue we have to define for each activity who are the responsible of the monitoring and the enforcement of the QoS. About the latter the service provider aims at defining all necessary strategies in order to maintain the QoS defined in the SLA. On the contrary with the monitoring, performed by the user or by a trusted third-party, possible QoS violation can be discovered.

5 Multi-channel Adaptation Strategies

As pointed out in Section 3, current proposals for service provisioning with QoS guarantees mainly focus on single-channel systems in which users can access a service exploiting a device and an underlying infrastructure. This enable designers to deploy architectures enforcing SLAs through mechanisms as resource monitoring, request/traffic admission control, prioritization, load balancing, and so on, essentially devoted to avoid performance degradation due, for example, to platform congestion and faults [13]. Analogous techniques are put in place to enforce other non-functional properties such as availability, and security (see Section 3). Even if these mechanisms are still necessary to enforce a quality level on each channel of a multi-channel service provider, the latter could miss opportunities if omitting to consider multi-channel adaptation strategies. Broadly speaking, these strategies are additional features enabling the system to dynamically react to asynchronous events notifying the modification of the conditions under which an SLA was issued, i.e. changes of user needs and of the underlying network and computing infrastructure states. The basic idea underlying these strategies is to react to these events by exploiting the availability of (i) several channels to provide the service and (ii) of the information on users available within the context and profile (see Section 2), in order to continue service provisioning without SLA violation.

Once an SLA has been established between a user and a provider, the service provisioning phase takes place. During this phase, however, according to the examples of Section 1, the user is allowed changing the channel for accessing the service, and channels can even modify their behavior possibly causing SLA violations. An adaptive system reacts by proposing to the user to switch onto another channel compatible with the user profile and the context.

To achieve such adaptiveness, it is first necessary to bridge the gap between low-level QoS properties and high-level QoS guarantees as expressed in SLA and SLS. This issue is clearly application dependent (e.g. translating frame rate, dimension and color depth in channel requirements as minimum bandwidth, maximum latency, and maximum jitter in the video-on-demand example). However, in the general case means to monitor low-level QoS parameters and to set thresholds to signal possible violations of SLAs must be put in place in order to enable applications to reason about service levels. The quality model presented in the previous section represents a useful framework (see Figure 4), as the channel, physical resource, and data resource models can be fed of values coming from a monitoring infrastructure (e.g. a JMX-based system [20], and the quality factory presented in [9]). Once the system is fed with values coming from resource monitoring tools, multi-channel adaptation strategies can be built on top of this knowledge base.

Such strategies can be implemented by enabling applications to handle two events, namely the user-triggered channel-switch, and the provider-triggered channel-switch, that model the condition under which a channel switch can occur.

In particular, a user-triggered channel-switch event is thrown by the user device to the provider upon detecting (i) a change of user preferences in terms of channels or devices to access a service or (ii) that the SLA has been violated. In the first case, in order to allow the user to switch among the available channels, a set of suitable operations which acts on the channel properties should be included in the service specification. For this matter, each service will have to provide operations as changeChannel(aChannel), which, given in input the channel definition, states the availability of the required channel with respect to the user context and eventually operates the channel switching. Moreover, in order to create a user-friendly system more expressive operations could be used as, for instance in video-on-demand example, playOnTV(), playOnPDA(). On the contrary, in order to discover SLA violations, the user can use its own devices able to fairly monitor service level, or indifferently can refer to trusted third party [19].

Provider-triggered channel-switch events are thrown to the
application when the usage of some resource exceeds a
given threshold, e.g. upon a channel exceeding a maximum
jitter value. To account for this, the quality model allows
setting thresholds on the metrics defined for low level re-
sources according to [19].
Upon the receipt of both user-triggered and provider-
triggered channel-switching events, the provider generates
a novel customized SLS taking by into account context, pro-
file, and available resources; then it suspends service provi-
sioning, sends the SLS to the user, and the system eventu-
ally come up with a new SLA, possibly after a negotiation
phase.

6 Conclusion and Future Work

In this paper we present a first proposal for novel qual-
ity model for multi-channel e-Services and proposing adap-
tation strategies that allow describing and enforcing
QoS aspects, respectively, developed in the context of the
MAIS project. In particular the quality model, which can
be viewed as an extension of existing proposals for
single-channel system [19], allows classifying resources of
a multichannel information systems, their properties
related to non-functional aspects, and defines the role of
the service provider as a proposer of service levels, and
of the user as a selector of the accepted service level.
Adaptation strategies allow providing users with service
levels that are compliant with the user equipment and
preferences, other than being implementable by service
providers. Furthermore, adaptation strategies enable the
system to dynamically react to asynchronous events as a
change of user preferences about channel exploitation or
a shortage of resources due to some unpredicted failure event.

As future work, first we plan to refine and to detail the qual-
ity model with the attributes related to the resources, es-
pecially those of channels\(^3\). Describing channels in terms
of QoS passes through the analysis of several attributes re-
lated to (i) physical devices (ii) data-link protocols, and (iii)
application protocols, and of their dependencies. The de-
scription of these attributes would allow feeding the model
with current values fetched from observable components
and thus to implement adaptation strategies involving con-
trollable components. Another issue requiring further in-
vestigation is the mapping between high-level QoS prop-
ties as expressed in an application dependent manner within
service levels and the low-level QoS properties, in order to
provide applications with possibly standard tools to easily
translate their service levels in resource-level requirements,
for at least a restricted set of applications, i.e. service levels.

Further investigations will be in the definition of the char-
acteristics of user profiles which translate to QoS require-
ments, such as for instance in the case for user with special
needs.

References

methodology for e-service substitutability in a virtual district
environment. In Conference on Advanced Information Sys-
tem Engineering (CAISE 2003), Klagenfurt-Velden, Austria,
CORBA Infrastructures. Software: Practice & Experience -
[5] A. Banerji, C. Bartolini, D. Beringer, V. Chopella, K. Govind-
darajan, A. Karp, H. Kuno, M. Lemon, G. Pogossiats,
S. Sharma, and S. Williams. Web Service Conversation Lan-
guage (WSCL) 1.0. W3C Web Site: www.w3.org/TR/
wscl10, March 2002.
[6] L. Baresi, D. Bianchini, V. De Antonellis, M. Fugini,
B. Pernici, and P. Plebani. Context-aware Composition of
e-Service. In Technologies for E-Services : Third Interna-
tional Workshop, TES 2003, Berlin, German, September 7-8,
2003.
compliant agent framework. In Proc. of the practical ap-
lications of intelligent agents and multi-agents (PAAM99),
[8] P. Brittenham, F. Cubera, D. Ehnebuske, and S. Gra-
ham. Understanding WSDL in a UDDI Registry. IBM
Web Site: www-106.ibm.com/developerworks/
[9] C. Cappiello, C. Francalanci, B. Pernici, P. Plebani,
and M. Scannapieco. Data Quality Assurance in Cooperative
Information Systems: a Multi-dimension Quality Certificate.
In International Workshop on Data Quality in Cooperative
Information Systems in conjunction with ICDT 2003, Siena,
Italy, January 10-11, 2003.
[10] E. Christensen, F. Curbera, G. Meredith, and S. Weera-
warana. Web Services Description Language (WSDL)
1.1. www.w3.org/TR/2001/NOTE-wsdl-20010315,
Web Services. www.daml.org/services/daml-s/
0.7/daml-s.html, October 2002.
[12] F. Curbera, Y. Goland, J. Klein, F. Leymann, D. Roller,
S. Thatte, and S. Weerawarana. Business Process Execution
Language for Web Services, version 1.0. www.ibm.com/


