A Quality Model for Multichannel Adaptive Information Systems

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

Barbara Pernici
Dipartimento di Elettronica ed Informazione – Politecnico di Milano
Piazza Leonardo da Vinci, 32
I-20133 Milano, Italy
barbara.pernici@polimi.it

Pierluigi Plebani
Dipartimento di Elettronica ed Informazione – Politecnico di Milano
Piazza Leonardo da Vinci, 32
I-20133 Milano, Italy
plebani@elet.polimi.it

ABSTRACT

The ongoing diffusion of novel and mobile devices offers new ways to provide services across a growing set of network technologies. As a consequence, traditional information systems evolve to multichannel systems in which services are provided through different channels, being a channel the abstraction of a device and a network. This work proposes a quality model suitable for capturing and reasoning about quality aspects of multichannel information systems. In particular, the model enables a clear separation of modeling aspects of services, networks, and devices. Further, it embeds rules enabling the evaluation of end-to-end quality, which can be used to select services according to the actual quality perceived by users.

Categories and Subject Descriptors

H.4.m [**Information Systems**]: Miscellaneous

General Terms

Design

Keywords

Quality of Service, Model

1. INTRODUCTION

The growing standardization, acceptance, and diffusion of novel devices (e.g., PDA, Smartphones) and network infrastructures (e.g. Wi-Fi, UMTS, Bluetooth) require an overall shift of the paradigms for design and development of traditional information systems in order to enable the access of services from users exploiting several distinct technologies. In other words, traditional information systems are evolving to *multichannel* information systems in which a service is provided on several distinct *channels*, being the latter an abstraction of the network infrastructure and of the device used to access a service. At a high level of abstraction, channel examples are: a simple PC connected to the service provider through the Internet, a PDA connected equipped with a wireless LAN network interface, a SmartPhone exploiting UMTS and a private backbone. The Italian MAIS project (Multichannel Adaptive Information Systems)[1] focuses on the study of models, methodologies, and tools for the design and development of this class of forthcoming systems.

In particular defining, providing, certifying, and monitoring quality of service in a multichannel information systems is one of the MAIS project objectives. Research on these topics is motivated by observing that in a multichannel information system the quality of a given service as perceived by users (i.e., *end-to-end QoS*) strictly depends on the channels used to access the service, i.e. on the quality of networks and devices, other than on the quality of the service itself. As an example, let us consider a Video-on-Demand (VoD) service, allowing users to watch a video stream using both “typical” devices and networks (e.g. a Pc connected to a wired network), as well as innovative wireless and mobile device (e.g., PDAs, SmartPhones)[2]. In this context, and especially for chargeable services, it is highly desirable to provide users with some guarantees about end-to-end QoS, in order to enable service selection and comparison not only on the basis of functional requirements, but also using information about non-functional aspects of the service. This requires service providers to carefully evaluate how the QoS they are able to offer on the basis of their internal resources is modified along the path connecting the user to the service itself. To this aim, we deem it necessary to develop a clean *quality model* that enables reasoning on how channels (and associated providers) affect non-functional aspects of a service. Once such a model is available, services can be labelled with end-to-end QoS information, thus enabling QoS-enhanced service offering (by providers) and service lookup (by users) in a multichannel adaptive information system. In addition, a quality model puts the basis for quality certification and monitoring. In particular, by clearly stating how each metric is affected by service, network, and device providers, the proposed model supports the definition of metrics, monitoring means, and responsibilities of service level enforcement within a service level agreement framework [27]. Finally, the model supports the design of user-side and provider-side channel adaptation strategies based on QoS [21], i.e., user criteria to trigger channel switches upon detecting unsatisfactory levels of quality while accessing the service from a given channel, or provider suggestions about channel switches for maximizing user quality, respectively.

In this paper we present a quality model for multichannel information systems. This model is composed by a *system model* and by a set of *roles* and *rules* that the elements of the system model respect and use to enable the eventual evaluation of end-to-end QoS.

[1] The project web site is available at http://www.mais-project.it

[2] Let us note that this class of “Mobile TV” services 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.

WWW2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

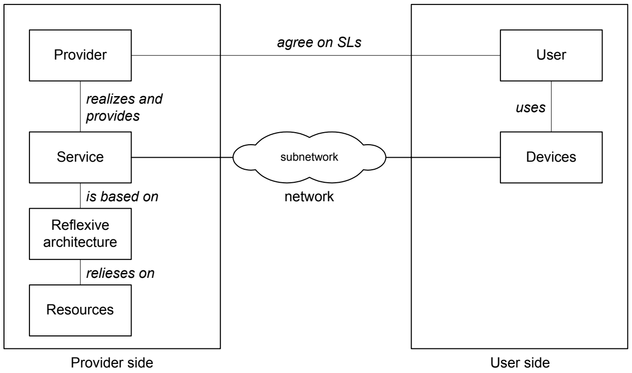

Figure 1: User perspective

In particular, in the system model we identify the main *objects* (services, networks, devices) and *actors* (service, network, and device providers) of multichannel information system. According to the model, actors coordinate their activities within *communities* to obtain *specifications*. The basic idea underlying the quality model is thus to extend specifications of objects with the quality information necessary to evaluate end-to-end QoS. Further, roles define who is responsible of defining quality parameters and quality sets for each object and each specification, while rules dictate how to combine quality sets associated to services, networks, and devices to evaluate end-to-end QoS. Finally, we show how the quality model supports QoS-enhanced service offering and lookup. We support the presentation using a running example, i.e., the implementation of a VoD service in a multichannel information system.

The remainder of this paper is structured as follows: Section 2 presents an overview of the MAIS project; Section 3 discusses related work; Section 4 illustrates the quality model; Section 5 sketches an application example; finally, Section 6 concludes the paper.

2. MAIS OVERVIEW

A preliminary analysis of the issues related to service provisioning in a multichannel environment straightly drives to the necessity of defining channels and their interactions with the two main actors it serves, i.e., the service provider and the service user. Figure 1 sketches our vision of service provisioning in MAIS, and points out the main issues we face.

Starting from the channel, using a simplified version of [23], it is defined as a pair $\langle network, device \rangle$, where *device* characterizes the user's equipment (e.g., PDA, PC, SmartPhone), and *network* identifies the set of devices interconnecting the provider and the user device.

Concerning the service provider, he is described in terms of the provided services. In particular, besides the traditional definition in which the performed functionalities are considered, a service is also defined by the set of channels used to carry the service information.

On the other channel side, the user is characterized by a *context* and by a *profile* [5]. The context describes, among the others, the set of channels available and the channel currently in use, in a given location and at a given time. The profile captures user preferences that depend 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 VoD 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 its preferences (defined in the profile), by stating, for instance, that when all the channels are available it prefers to use the PC when it stays at home.

The service provider and the service user, before the service provisioning starts, have to agree on a contract concerning the quality of the provided services. In particular, the contract should define which is the quality level of the service provided in terms of availability, data quality, security and so on. Other entries of the contract should define which are the available channels through which the service may be invoked, or which is the device the user is going to use.

During service execution, the service provider is in charge of maintaining what stated in the contract. For this matter, it relies on monitoring strategies and adaptivity policies that are able to discover contract unfulfillment and to consequently enact the suitable adaptation policies.

In order to allow for adaptivity from the provider standpoint, services are provided using a reflective architecture [18] that handles the underlying resources in order to enabling the suitable policies to maintain a pre-defined service level. On the contrary, from a user standpoint, the adaptation is not motivated by the contract but by his own requirements with respect to his profile.

3. RELATED WORK

Quality of service and related issues have been the topic of several researches and standardization efforts crossing distinct communities during the last years. For the purposes of this work, it is worth mentioning the web and web-service community (e.g., [25, 20, 29]), the networking and internetworking communities (e.g., [12, 13, 7, 11]) as well as the middleware community (e.g., [30, 22]). Even if different in nature, the objectives of these efforts were at least intended (i) to identify the relevant *measurable* characteristics affecting the quality of the services provided by a given "object" (e.g., a web-service, a network infrastructure, a middleware platform) and (ii) to define means (e.g. architectures, paradigms, components, and protocols) to implement an "object" whose values of its measurable characteristics satisfy some quantitative constraints. We name the measurable characteristics of objects *QoS parameters*[3].

For instance, focusing on the web-service community, [20] identifies the QoS parameters deemed useful for service providers to characterize the services. In some sense, [25] integrates this proposal devising means to extend service discovery on the basis of QoS-related information in addition to interfaces. [29] proposes a methodology enabling the evaluation of the overall QoS of a composite service, i.e., a service obtained by composing several distinct services, provided that a description of the QoS parameters of the component services is available. Unfortunately, several of such parameters (e.g., availability, performance, and so on) are difficult to define and to evaluate in an end-to-end manner since the networks or the devices used to let a service and users interact are even partially out of the control of the provider. For this matter, in these proposals services are labelled by providers with a set of parameters in order to enable ordering and selecting them on the basis of QoS. This approach is limited when considering that usually the QoS perceived by users depends on factors out of the control of the provider, e.g., service availability depends on (i) provider availability, (ii) network availability, and (iii) user device availability.

To address this kind of issues, [28] introduces the relevant dis-

[3] Several other names have been proposed in the literature on the topic, e.g. *dimension*, *attributes* [9].

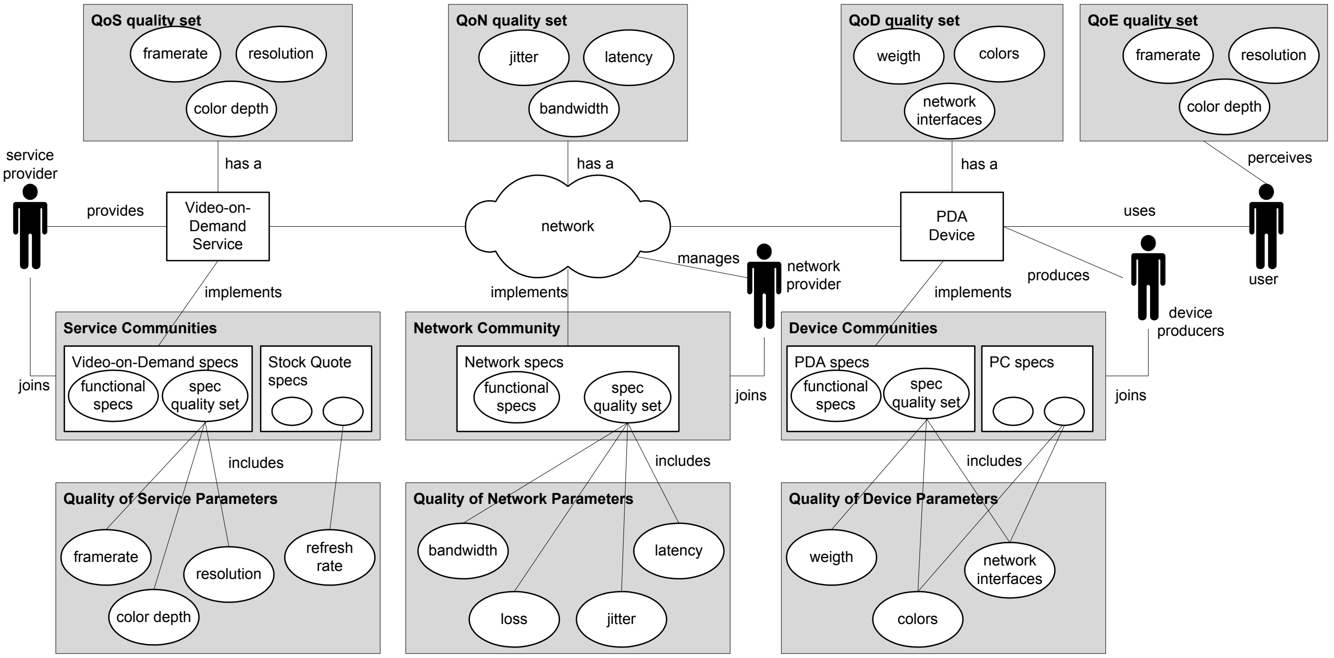

Figure 2: The system model of the quality model

tinction between Quality of Experience (QoE) and QoS in the web environment, by pointing out that while QoS parameters are under full control of the service provider (e.g., throughput, server availability), QoE parameters, even if closely related to QoS parameter, may be influenced (i) by subjective elements related to user history and preferences, and (ii) by *any* system interposed between the provider and the user. Evidences of this important distinction are presented in [17], experimentally showing that the QoE of a user surfing the web using a browser through an Internet Service Provider is lightly affected by network latency and highly affected by network bandwidth. Similar work, e.g., [26], show how to relate QoS of service and network providers to QoE for *specific* services and applications. In other words, QoE can be expressed as the user perception of end-to-end QoS [3]. Putting the basis for evaluating QoE of services is a first class issue, as it would allow coping quite straightforwardly with other relevant issues such as (i) the definition of *service level agreements* [27, 16] based on the actual quality of service perceived by users, (ii) the clear and unambiguous assignment of responsibilities for enforcing each quality parameter value (and to possibly define the corresponding penalties in case of unsatisfactory levels) [24, 14], and (ii) the definition of integrated architectures among service, network, and device providers, cooperating to enforce QoE parameters [19, 15].

4. THE QUALITY MODEL

The quality model consists of (i) a *system model* defining objects and actors, and of (ii) a set of roles and rules enabling the association of quality information to objects. Quality information is expressed using *quality parameters* and *quality sets*.

4.1 System Model

The proposed system model is based on the service oriented architecture (SOA, [10, 4]), which is extended with an explicit representation of channels (i.e, networks and devices) to fit multichannel information systems. As a consequence, the SOA basic infrastructure based on service providers, service requestors, and service directories, is augmented with networks, devices, and associated actors. Figure 2 illustrates the main concepts of the resulting systems model, consisting of *objects*, *actors*, *communities*, and *specifications*, described below.

Objects. *Services*, *networks*, and *devices* are the first class objects of our system model. They are handled by actors and specified by communities, i.e., an object is the implementation of a specification defined by some community (see below). In particular, a *service* is a software entity performing operations that can be characterized by a service specification defining its functionalities. Several proposals address the problem of expressing such a *functional specification* through a standard language (e.g., WSDL [6], DAML-S [8]). A *network* abstracts the set of devices used to transport messages between a service provider and the user device, which may include several subnetworks, routers, gateways, and other devices and could be characterized by the interface offered to devices (e.g., 802.3, 802.11, GPRS, UMTS) and by the set of users and providers that may connect to the network. A *device* is the user mean for exploiting services, and it consists of a set of input and output interfaces that enable interactions with users (e.g., keyboard, display), and by a set of network interfaces (e.g., supporting 802.3, 802.11, GPRS, UMTS) enabling the connection to a network.

Actors. Actors are entities responsible of performing actions on objects or interested in the execution of some operations. In particular, according to the service-oriented architecture [10], *service providers* publish the specifications of the provided services within a public registry (e.g., a UDDI registry [2]), maintained by the *service directory*. *Network providers* offer networking services and manage the underlying network infrastructure, e.g., cellular phone companies, Internet Service Providers (ISP). *Device producers* model industries producing devices as PDAs, PCs, SmartPhones, and so on. *Users*, using the service directory, are able to select, to locate, and to invoke services by analyzing functional specifications.

Communities and specifications. Standardization efforts, carried out by organizations and consortia, result in specifications. Therefore a community is a group of actors which aims at proposing a specification for a group of objects with some relevant common characteristics. An actor in the community implements objects compliant to the specification defined by the community. In particular, service providers standardize the minimum functional specification a service must implement to be included into a particular *service community* (e.g., the VoD community). In other words, a service community can be characterized by a minimum service specification. Network communities define the main characteristics of networks through network specifications. Analogously, providers of similar devices identify a minimum set of features (e.g., display size, color depth, audio capabilities, and network interfaces) that members of a *device community* must support.

As mentioned above, the quality model must enable association of quality information to objects. This is achieved extending specifications with quality information using quality parameters, sets and rules for composing them, described below.

4.2 Quality parameters, sets, and rules

A quality parameter models relevant and measurable non-functional aspects of an object. Quality sets are compounds of quality parameters that can be associated to objects. Quality rules define how to combine quality sets associated to distinct objects to evaluate their global effects on their non-functional aspects. In the following, we introduce the notation deemed necessary to deal with these concepts.

Quality parameters. A quality parameter qp is a pair

$$qp = \langle name, admissible\_values\rangle$$

where $name$ identifies the parameter and $admissible\_values$ is an *ordered* set of typed values within which the parameter is admitted to vary. Examples of quality parameters are:

$\langle bandwidth, [1Kps..512Kps]\rangle$

$\langle encryption, [40bit; 64bit; 128bit]\rangle$

$\langle resolution, [320 \times 200; 800 \times 600; 1024 \times 768; 1240 \times 748]\rangle$,

$\langle latency, [10ms...500ms]\rangle$.

Given a quality parameter qp, $qp.name$ returns the name field, and $qp.values$ returns the admissible values. Further, the functions $best(name)$ and $worst(name)$ are introduced to evaluate the admissible value corresponding to the best and the worst quality of a quality parameter $qp : qp.name = name$, respectively. The values of these functions clearly depends on the semantics of the parameter, e.g. $best(bandwidth) = 512Kbps$, $best(latency) = 10ms$.

Quality sets. Quality sets are classes of quality parameters. Given a quality set QS, $QS.names$ returns the set of names of the quality parameters belonging to q. We say that a quality set QS_2 is *contained* within a quality set QS_1 ($QS_1 \subseteq QS_2$) *iff* (i) QS_1 and QS_2 have the same set of names, i.e. $QS_1.names = QS_2.names$, and (ii) each quality parameter in Q_1 has admissible values ranging within the admissible values of the corresponding parameter in Q_2, i.e. $\forall qp_1 \in QS_1, qp_2 \in QS_2, qp_1.name = qp_2.name \Rightarrow qp_1.values \subseteq qp_2.values$. Quality sets can be associated to both *objects* and *specifications*. Therefore, provided that object O is the implementation of a specification S, we distinguish between *specification quality sets* ($S.QS$) and *object quality sets* ($O.QS$). Using $S.QS$ a community defines relevant quality parameters and their admissible values. These values represent all the possible values that any object implementing the specification can offer. As a consequence we require that if O implements (or is compliant to) S, then $O.QS$ is contained within $S.QS$. Figure 3 illustrates the effect of containment between quality sets on the framerate quality parameter of a VoD service (expressed in frame per seconds - fps).

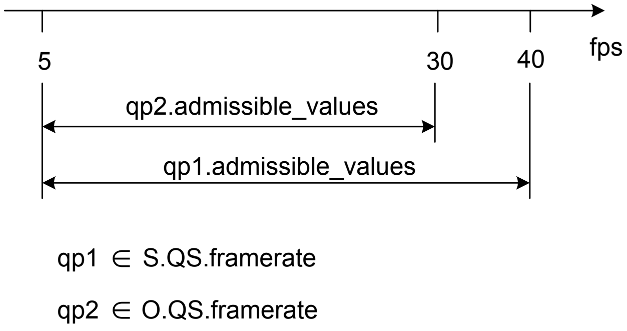

Figure 3: Example of the effect of containment between quality sets

Quality rules. Actors and communities first associate quality parameters and sets to objects and specifications, and then they relate quality sets using *quality rules*. A quality rule can be defined as follows: given two quality sets QS_1, QS_2, a quality rule for QS_1 *on* QS_2 is a function $qr_{12}(QS_1, QS_2) = QS_3$ such that: (i) QS_3 is a quality set, and (ii) QS_3 is *contained* in QS_1. This implies that independently from the set of quality parameters of QS_2, QS_3 has parameters whose names are the same of those contained in QS_1 and whose values are contained in the corresponding parameters of QS_1. Indeed, these rules are used to model the effects on the quality parameters of an object/specification A having a quality set QS_1 when it is coupled with another object/specification B having a quality set QS_2, i.e., to evaluate how the quality of A is affected by B. Further, we require that quality rules satisfy the following *containment property*: if $qr_{12}(QS_1, QS_2) = QS_3$ is a quality rule, then for any $QS'_1, QS'_2 : QS'_1 \subseteq QS_1, QS'_2 \subseteq QS_2$, $qr_{12}(QS'_1, QS'_2) = QS'_3$ is a quality rule, and $QS'_3 \subseteq QS_3$. It is easy to see that the quality rule containment property implies that if a quality rule is defined among specification quality sets, it can be used to evaluate effects on quality of the corresponding object implementations.

Remarks. The main objective of the quality model is to analyze how the relationships among the elements of a multichannel information system affect the quality perceived by users. To this aim, the containment relationship forces actors to implement objects with quality attributes compliant to the specification, while quality rules (which have a similar containment relationship) dictate how to evaluate quality modifications of a service upon varying the channel used to access the service itself. Further, being the focus on relationships, we do not enter the details of a language for specifying quality parameters and sets, and we limit to mention a good candidate, i.e., the Quality Modelling Language (QML, [9]).

4.3 Roles for quality management

Exploiting quality parameters, sets, and rules communities and actors operate in two different stages, namely the *specification definition phase* and the *object implementation phase*, in order to specify quality aspects of objects and specifications and to analyze their relationships. In particular, first communities define the specification quality sets for services, networks, and devices. Once these information are available, during the object implementation phase, each actor can produce objects compliant to specifications according to the containment relationships, can define quality rules enabling to evaluate the end-to-end quality of a service for each channel, and in particular service providers can evaluate information on the end-to-end quality of service as a function of the channel on which the service is provided.

Communities roles. During the specification definition phase, communities are in charge of defining quality specifications (i.e., quality sets). In particular, communities identify the name and the admissible values of all the quality attributes identifying all the relevant measurable non-functional aspects of any object that will comply to the specification. In particular, (i) each service community associates to each service the service specification quality set ($S.QS_{QoS}$), (ii) each network community associates to each network the network specification quality set ($S.QS_{QoN}$), and (iii) each device community associates to each device the device specification quality set ($S.QS_{QoD}$). Examples of these specifications can be found in [13, 7] for networks, and in [1] for devices. Concerning services, we remark that current proposals, e.g., [25, 29, 24], commonly model non-functional aspects of services (e.g. performance, availability, and so on), assuming either that they can be evaluated almost independently from networks and devices or including the end-to-end effects of a single channel within the service quality set specification. In both cases, they result not suitable for multichannel information systems. It is also important to note that quality parameters in the service specification quality set shall not consider channels and end-to-end aspects. For example, the service response time should be the time elapsed to produce a reply since the arrival of a request at the provider site; analogously, service availability should be evaluated at the provider site without considering network and device effects. This feature highly simplifies the definition of these service specifications. For each quality parameter identified, the communities set the limits of $admissible\_value$ to values which include all the possible values independently from the implementation. When precise values cannot be identified, both or one of the two $admissible\_value$ limits can be set to $\pm\infty$. For example, the VoD community, according to the video encoding softwares now available, can set the $admissible\_value$ of the framerate quality parameter in the VoD specification to $[5fps..40fps]$. On the contrary, considering the latency, the network community define the $admissible\_value$ as $[0sec.. + \infty]$ since some implementation could be follow a best effort approach about this parameter. Finally, a service community can relate the specification quality set $S.QS_{QoS}$ of a service with specification S to a set of channels. To attain this, for each channel $c_i = \langle N, D\rangle$, the community defines a quality rule

$$qr_{s,c_i}(S.QS_{QoS}, qr_{n,d}(S.QS_{QoN}, S.QD_{QoD}))$$

Due to the quality rule containment property and to the quality set containment relationship, such a quality rule enables every service provider, aiming to offer a service with specification S on channel c_i, to evaluate the end-to-end QoS. This evaluation is obtained by simply substituting the object quality sets $O.QS_{QoS}$, $O.QS_{QoN}$, and $O.QS_{QoD}$ within the rule, provided that object quality sets are contained within those of the corresponding specification, i.e. $O.QS_{QoS} \subseteq S.QS_{QoS}$, $O.QS_{QoN} \subseteq S.QS_{QoN}$, and $O.QS_{QoD} \subseteq S.QS_{QoD}$. In order to define quality rules, communities exploit engineering knowledge, e.g., physical dependencies among quality parameters. As example, the bandwidth parameter of networks significantly affects the framerate, color depth and resolution quality parameters of a VoD service which are also affected by the colors quality parameters of a device.

Actors roles. During the object implementation phase, actors implement objects compliant to specifications and associate them object quality sets that satisfy containment relationships with the corresponding specification. Adhering to specifications is important to let service providers evaluate end-to-end QoS using (i) their object quality sets, and (ii) the set of quality rules dictated by the service community (if any) as described above. Let us note that "adhering to specifications" in our vision is equivalent to say that object quality sets are included in specification quality sets. To attain this, it is enough to let values of quality parameters of specification quality sets vary in interval sufficiently large for any possible implementation. Note also that if there is no quality rule for given a channel on which the provider aims to offer a service, the actor is free to define a quality rule for the channel and to propose it to the service community for approval. Therefore we can assume that exploiting the proposed model, service provider are eventually able to label service with the quality that will be perceived by users accessing the services from all the channels over which the service is provided.

Remarks. A relevant issue about the proposed quality model concerns the quality of its instances, this meaning answering the following question: "How good is this instance of the quality model in capturing quality aspects of this particular multichannel information system?". Answering this question relates to the attention paid by communities and actors in defining quality sets and quality rules. In this sense, the proposed model is a first step whose main contribution is in *enabling* the creation of good instances by a clear definition of the underlying system model, tools and responsibilities. The process to build instances of the proposed model is applied to a case-study in the following section.

5. THE VOD EXAMPLE

The focus of this section is to sketch the use of the quality model presented in this paper mainly in order to clarify the phases and tools introduced. Figure 2 shows all the main objects and actors used in this example. As aforementioned, the Video-on-Demand is a service that allows a user to watch to video-streams on several heterogeneous devices, e.g. traditional PCs and SmartPhones. Let us remark that this service has been recently introduced in the European mobile phones marketplace and currently provides users with short streams containing relevant pieces of football matches. Therefore, in this simple example, the communities involved are: the VoD service community (SC), the network community NC, and two device communities (PCC for the PCs and SMC for the SmartPhones).

First of all, recall that the model assumes that communities define the functional specifications and identify the set of quality parameters deemed relevant with respect to the services they provide. Therefore we assume communities to define the following quality sets.

SC defines the quality set $VoD_{spec}.QS_{QoS}$ as follows:

$VoD_{spec}.QS_{QoS} = \{\langle framerate, [5fps..40fps]\rangle,$
$\langle colordepth, [2bit..24bit]\rangle,$
$\langle resolution, [320 \times 200; 800 \times 600; 1024 \times 768]\rangle\}$

NC defines the quality set $MyNet_{spec}.QS_{QoN}$ as follows:

$MyNet_{spec}.QS_{QoN} = \{\langle bandwidth, [10Kbps...512Kbps]\rangle,$
$\langle loss, [0..0.01]\rangle,$
$\langle latency, [5ms..500ms]\rangle\}$

PCC defines the quality set $Pc_{spec}.QS_{QoD}$ as follows:

$Pc_{spec}.QS_{QoD} = \{\langle colors, [2..24bit]\rangle,$
$\langle nw\_int, [802.11b; 802.3]\rangle\}$

SMC defines the quality set $SmartPhone_{spec}.QS_{QoD}$ as follows:

$SmartPhone_{spec}.QS_{QoD} = \{\langle weight, [150gr..300gr]\rangle,$
$\langle colors, [2..16bit]\rangle,$
$\langle nw\_int, [GPRS; UMTS; GSM]\rangle\}$

Communities are also responsible of defining the quality rules relating quality parameters. These rules originates from the analysis of the physical dependencies among the service, network, and device parameters. Considering the VoD example, and considering very simplified versions of the actual mathematical relationships, it is possible to state that:

$$framerate \star colordepth \star resolution = K \star bandwidth$$

According to this relationship we can also identify the following relationships which affects the admissible range of the quality parameters.

$$best(framerate) = \frac{best(bandwidth)}{worst(colordepth) \star worst(resolution)}$$

$$worst(framerate) = \frac{worst(bandwidth)}{best(colordepth) \star best(resolution)}$$

$$best(colordepth) = \\ minimum(best(colors), best(colordepth))$$

$$worst(qoe.colordepth) = \\ minimum(worst(colors), worst(colordepth))$$

With similar reasonings, it is possible to derive how each service quality parameter is affected by (i) channel quality parameters and (ii) device quality parameters, i.e. to define quality rules. Therefore we abstract out from the detailed sequence of steps taking to the precise derivation of quality rules, which are the main objectives of some ongoing works. We limit to mention that observing mathematical relationships as those described above, it is possible to precisely characterize these rules. In particular, it is possible to evaluate two end-to-end quality rules, namely $qr_1()$ and $qr_2()$, that take a service quality set as input parameter and return the service quality perceived by users accessing the service through network $MyNet$ and using either a PC or a smartphone, respectively. Let us now suppose that four different actors (namely, a service provider S_1, a network provider N_1, a PC provider D_1, and a smartphone provider D_2) implement all the specifications described above. According to actor roles, besides the implementation of the functional specification, actors are in charge of defining an object quality set that satisfies the containment property with respect to related specification. Therefore we assume these providers to define the following object quality sets.
S_1 defines the $S_1.QS_{QoS}$ quality set as follows:

$S_1.QS_{QoS} = \{\langle framerate, [5fps..30fps]\rangle,$
$\langle colordepth, [2bit..24bit]\rangle,$
$\langle resolution, [320 \times 200; 800 \times 600]\rangle\}$

N_1 defines the $N_1.QS_{QoN}$ quality set as follows:

$N_1.QS_{QoN} = \{\langle bandwidth, [10Kbps...128Kbps]\rangle,$
$\langle loss, [0..0.01]\rangle,$
$\langle latency, [10ms..500ms]\rangle\}$

D_1 defines the $D_1.QS_{QoD}$ quality set as follows:

$D_1.QS_{QoD} = \{\langle colors, [2..24bit]\rangle,$
$\langle nw\_int, [802.3]\rangle\}$

D_2 defines the $D_2.QS_{QoD}$ quality set as follows:

$D_2.SmartPhone_{spec}.QS_{QoD} = \{\langle weight, [170gr..180gr]\rangle,$
$\langle colors, [2..8bit]\rangle,$
$\langle nw\_int, [UMTS]\rangle\}$

These specifications, together with the quality rules qr_1 and qr_2, allow the user to derive the end-to-end quality of service for each available channel. Note that, in this simple example, the admissible channels are two (obtained combining network $MyNet$ with the devices D_1 and D_2). For each of these channels the user can evaluate how the service quality set $S_1.QS_{QoS}$ is perceived through these channels applying the quality functions. Such an evaluation generates, for each channel, a new quality set $U_i.QS_{QoS} \subseteq S_1.QS_{QoS}$ (i identifies the channel). Note that $U_i.QS_{QoS}$ is composed by the same quality parameters belonging to $S_1.QS_{QoS}$ with an $admissible\_range$ restricted if necessary. Let us also finally remark that the same evaluation can be performed by service providers in order to provide users with a precise characterization of the actual end-to-end quality of service they are going to perceive upon accessing the service. This feature can be used perform service composition basing on dynamic, end-to-end quality of service specifications that depend on the overall path followed by service information.

6. CONCLUDING REMARKS

In this paper we have presented a quality model for multichannel information systems. In this context, and in particular if we follow an adaptive perspective as the one adopted in the MAIS project, service provisioning and its associated quality values cannot be automatically determined. In fact, channels and user requirements can change during or before service provisioning, and on the other hand the service provider can dynamically improve the provisioning quality if necessary.

To enable such an adaptation, the model takes into account the different objects which compose a multichannel information system, i.e., service, network, device, and user. Each of these objects is modeled with respect to the related community which defines the quality parameters and their admissible ranges. Referring to the community, in fact, we are able to normalize the description of the objects in order to simplify the comparison among them.

The proposed model is therefore a general framework which is useful for both the contract specification and contract enactment. In the first case the quality parameters can be used to specify multiparty contracts between all involved partners, defining admissible values for each involved object. Once the contract is defined, the quality parameters are also used to monitor potential problems and to adapt service provisioning both on provider side and on channel side to context characteristics changed during the enactment.

7. ACKNOWLEDGMENTS

Part of this work has been supported by the Italian MIUR FIRB MAIS Project.

8. REFERENCES

[1] Distributed Management Task Force Standards. http://www.dmtf.org/standards.

[2] Uddi Web Site. http://www.uddi.org/.

[3] *Workshop on "End-to-End Quality of Service. What is it? How do we get it?"*, ITU Headquarters, Monbrillant Building, October 2003. http://www.itu.int/ITU-T/worksem/qos/.

[4] ACM. Communications of the ACM, October 2003.

[5] L. Baresi, D. Bianchini, V. D. Antonellis, M. G. Fugini, B. Pernici, and P. Plebani. Context-aware Composition of e-Service. In *Technologies for E-Services : Third International Workshop, TES 2003, Berlin, German, September 7-8, 2003*.

[6] R. Chinnici, M. Gudgin, J. Moreau, and S. Weerawarana. Web Services Description Language (WSDL) 1.2. http://www.w3.org/TR/2003/WD-wsdl12-20030124, January 2003.

[7] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick. A framework for QoS-based Routing in the Internet. Internet Engineering Task Force - RFC 2386, August 1998.

[8] DAML Service Coalition. DAML-S: Semantic Markup For Web Services. http://www.daml.org/services/daml-s/0.7/daml-s.html, October 2002.

[9] S. Frølung and J. Koistinen. QML: A Language for Quality of Service Specification. Technical Report HPL98-10, HP Labs, HP Software Technologies Laboratory, February 1998.

[10] H. HP. Web Services concepts A technical overview. http://www.bluestone.com/downloads/pdf/web_services_tech_overvie%w.pdf.

[11] G. Huston. Next Steps for the IP QoS Architecture. Internet Engineering Task Force - RFC 2990, November 2000.

[12] I. (ITU). Terms and definitions related to quality of service and network performance including dependability. ITU Recommendation E.800 (08/94), August 1994.

[13] I. (ITU). Communications Quality of Service: A framework and definitions. ITU Recommendation G.1000 (11/01), November 2001.

[14] L. Jin, V. Machiraju, and A. Sahai. Analysis on Service Level Agreement of Web Services. Technical Report HPL-2002-180, HP Labs, June 2002.

[15] G. Kar, A. Keller, and S. Calo. Managing Application Services over Service Provider Networks: Architecture and Dependency Analysis. In *Proc. of the 7th IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2000.

[16] A. Keller and H. Ludwig. 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, May 2002.

[17] S. Khirman and P. Henriksen. Relationship Between Quality-of-Service and Quality-of-Experience for Public Internet Service. In *In Proc. of the 3rd Workshop on Passive and Active Measurement*, Fort Collins, Colorado, USA, March 2002. http://www.labs.agilent.com/hosted/conferences/pam2002/proceedin%gs/Relationship_Between_QoS_and_QoE.pdf.

[18] F. Kon, F. Costa, G. Blair, and R. H. Campbell. The case for reflective middleware. *Communications of the ACM*, 45(6):33–38, June 2002.

[19] V. Machiraju, A.Sahai, and A. P. A. van Moorsel. Web Services Management Network: An Overlay Network for Federated Service Management. In *Proc. of the Eighth International Symposium on Integrated Network Management*, pages 351–364, March 2003.

[20] A. Mani and A. Magarajan. Understanding quality of service of your Web services. *IBM Developer Works*, January 2002. http://www.ibm.com/developerworks/library/ws-quality.html.

[21] C. Marchetti, B. Pernici, and P. Plebani. A quality model for e-Service based multi-channel adaptive information systems. 1st Web Services Quality Workshop (WQW 2003) in conjunction with WISE Conference 2003, Rome, December 2003.

[22] C. Marchetti, A. Virgillito, and R. Baldoni. Enhancing Availability of Cooperative Applications through Interoperable Middleware. *Journal of Information Science and Engineering*, 19(1):39–57, January 2003. special issue on parallel and distributed systems.

[23] A. Maurino, B. Pernici, and F. A. Schreiber. Adaptive channel behaviour in financial information system. In *UMICS Workshop in conjunction with CAISE 2003, Klagenfurt-Velden, Austria, June 16-17, 2003*.

[24] J. M. Myerson. Guarantee your Web Service with an SLA. *IBM Developer Works*, April 2002. http://www.ibm.com/developerworks/library/ws-sla.

[25] S. Ran. A model for web services discovery with QoS. *ACM SIGecom Exchanges*, 4(1):1–10, Spring 2003.

[26] M. Siller and J. Woods. Improving Quality of Experience for Multimedia Services by QoS arbitration on QoE Framework. In *in Proc. of the 13th Packed Video Workshop 2003*, Nantes, France, April 2003. http://www.polytech.univ-nantes.fr/pv2003/papers/pv/papers/cr104%7.pdf.

[27] R. Sturm, W. Morris, and M. Jander. *Foundations of Service Level Management*. SAMS, 2000.

[28] A. van Moorsel. Metrics for the Intenet Age: Quality of Experience and Quality of Business. Technical report, HP Labs, July 2001. appears also in the proceedings of the 5th Performability Workshop, Spetember 16, 2001, Erlagen, Germany.

[29] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Z. Sheng. Quality driven web services composition. In *Proceedings of the twelfth international conference on World Wide Web*, pages 411–421. ACM Press, 2003.

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