

# Quality of service negotiation based on device capabilities and user preferences

K. El-Khatib, X. He, G. v. Bochmann

School of Information Technology & Engineering,  
University of Ottawa, Colonel By Hall  
(A510) P.O. Box 450 Stn A, Ottawa, Ont., Canada  
K1N 6N5  
Email: {elkhatib,xhe,bochmann}@site.uottawa.ca

**Abstract:** Most existing multimedia applications require the user to select certain QoS parameters. Applications are generally inelastic concerning the media types they can process and present to their users. But as the range of hardware, software, and media types is getting wider, the need for an automated system to do the selection process is increasing. In this paper, we developed a scheme for representing and storing the user preferences in a user profile. Applications may use this user profile to customize the presentation of the media content to the user's preferences. The device capabilities and capacities are also stored in a device profile. The design of a QoS aware middleware (user agent) is presented. On behalf of its user, the middleware merges the user and device profiles for all participants to make the selection of QoS parameters. The middleware will also allow the device alignment to enable the establishment of communication session. The architecture of the middleware will also ease the introduction of new services from various service providers.

Keywords: QoS, User QoS, User Profile, Device Profile, User Satisfaction, Device Capabilities, User Preferences, Agent, Service, Service Component, Middleware.

## 1. Introduction

With the bandwidth becoming so abundant at the core of the network (Gigabit Ethernet, DWDM,..), research attention is now shifting from managing network resources to managing end-user requirements. Powerful and diverse end-user devices are getting more and more available, and many new sophisticated multimedia services are becoming widely available. Moreover, most of the service providers are looking into ways to provide services that are much more tailored to the demand of end-user than the old traditional services (text based). While several emerging network technologies [1,2,3] are aiming to provide different Quality of Services (QoS) at the network level, most end users have little or no knowledge of these advances, and using them is considered too technical. For most end users, the importance is to get a required quality at a specified maximum price. End-users would prefer to specify their requirements and preferences regarding the quality and pricing of a service, and their end system should find out the best offer from service providers to satisfy these requirements and preferences. The end user will specify all of his preferences in his profile, and this profile will be available later on for the system to make an automatic selection of session parameters. The users should also be able to specify what type of service they would like to have when communicating with every specific person. A QoS-aware end-system should select the configuration parameters for the communication session depending on the capabilities of the end devices and the user's preferences. This selection should cover the limitations of all participants' software and hardware components. Based on the user's requirements, the end system might need to download and install new services in order to meet the requirements of the user.

In this paper, we show how a QoS aware end-system agent makes, on behalf of its user, an automated selection of the hardware and software components, media types, and configuration parameters for a teleconferencing session. This selection is based on: (a) hardware and software limitations on the used devices (b) personal preferences concerning individual media parameters and (c) the end-users preferences concerning the choice of media types and the price to pay. The architecture of the agent is an open architecture that allows easy and simple introduction of new services and functionalities to the system. This architecture speeds up the process of introducing new services into the network.

The rest of the paper is organized as follows: Section 2 presents an example scenario for a phone-conference between 2 end-users. In Section 3, we high-light the major steps in the establishment of the phone-conference. Section 4 presents an architecture for an open middleware that enables dynamic configuration and ease introduction of services. Section 5 gives a literature review for some existing protocols that are used to locate service in intranets and the Internet.

In Section 6, we present a survey of related work done on device profiles and automatic QoS selection. Then, in Section 7, we talk about the problem of QoS negotiation for video telephony and conferencing and our approach to solve it. Section 8 talks about the user profile and its possible representation and storage. In Section 9, we elaborate on how the device and user profiles can be used to make automatic QoS parameter selection. Section 10 continues with the example scenario and finally, Section 11 concludes the paper.

## 2. Example Scenario

Consider the following scenario: Alice, the user of terminal A, wants to invite Bob to a phone-conference session, using audio and video if possible. The first step is to find where Bob is currently located. Using some directory service (SIP Registrar, Contact Agent [14]), Alice's terminal finds the current position of Bob: Bob is currently using terminal B. On behalf of Alice, terminal A sends an invitation to terminal B. The invitation message will include the identity of the inviter, a proposal for the type of communication session (phone conversation, teleconference, white board) and the charging policy (Alice will cover the charge for the session). Acting on the behalf of Bob, terminal B checks into the profile of Bob to see whether Bob accepts calls from Alice. In his profile, Bob has already indicated that he is always willing to talk with Alice using the audio and video communication. Bob's profile contains also his preferences for the quality of communication media (TV quality for the video and telephone quality for the audio). Terminal B knows that it has a camera, microphone and speakers attached to it that enable this type of communication. Terminal B sends back to A the filtered profile for its hardware/software tailored to the preferences of Bob. Terminal B might notify Bob that Alice is trying to contact him, and he has to confirm the call acceptance before it signals back terminal A. Terminal A also builds its hardware/software profile aligned with Alice's preferences to communicate with Bob. After examining B's profile, terminal A determines that terminal B does not have an audio codec that is compatible with its own codecs. Terminal A sends to terminal B a message suggesting the URL where the codec could be found. Terminal A might only suggest to terminal B that it needs to download a specific codec before starting the session. Terminal B will try to find the codec in his domain first, and he might search the Internet for it. Before Downloading the codec, terminal B might consult with Bob in case there are some charges for the download of the codecs or in order to check the authenticity of the codec software. After downloading the codec, terminal B sends an update message for its profile to terminal A. Terminal A knows now that it is now aligned with terminal B, and the second step is to decide on the parameters for the codec, and more specifically on the QoS parameters.

Both Alice's and Bob's profiles include information about individual QoS parameters, but terminal A's agent has to find the exact values within the acceptance range of both Alice and Bob. Selection of these parameters is subject to several constraints including: devices capabilities, user preferences (frame rate, resolution), charging prices, etc... In case there was a common set of parameter values, terminal A will generate the transport requirements for the session (bandwidth, throughput, delay, jitter, loss rate, etc..). These transport requirements are passed to a bandwidth broker that signals the network for the establishment of the transport session. If this was successful, terminal A launches the

software (application) needed for the session. Terminal A also sends a message to terminal B containing the communication parameters for the session. Terminal B will also launch the required application with the selected parameters. Alice and Bob are now ready to carry their conversation. When Alice and Bob are through, Alice (or Bob) instructs her terminal to terminate the session. Terminal A or B (depending on what was agreed on) receives the electronic bill from the bandwidth provider covering the charges for the session.

A variation to the scenario would be that Alice's or Bob's profile indicates that the session should be secure. Both terminals have to decide on a security session key. Both terminals may use the service of a third party to select a session key. Note that the parties' authentication should be included in the contact phase of the session.

Another variation would be that Alice and Bob decided to invite a third person to the session. An additional service would be needed here: floor control. Floor control is needed for the coordination between all participants in the session. It is also a requirement for the sharing of resources (audio channel, whiteboard,...) without conflict.

### 3. Analysis of the Example Scenario

By examining the previous scenario, one can identify major points or steps; other scenarios may require minor variations to these steps:

- (a) **Locating the called party:** The problem of finding the current device on which the called party could be contacted at the time of the session initiation is a significant problem. The choices of "being connected" are increasing with many new technologies: mobile phone, home/office phone, PC at home or office, pager, etc... Many technologies are now evolving around the idea of keeping track of the location of a person as to which device he/she can access at this time. The "call forwarding" and "follow me" services in the PSTN domain are just few solutions to the problem. SIP Register and Home Directory [14] are other evolving suggestions for the Internet world.
- (b) **Locating the user profile:** Other than locating the currently used device, there is also the problem of locating the party profile and also the device profile. While it is natural to store the device profile on the device itself, the user profile has to be stored in a place that is accessible from any access point. [14] talks about the concept of building a Home Directory where the user profile can be stored and accessed from anywhere on the Internet.
- (c) **Specification, filtering and merging of profiles:** While we believe that the user preferences should be easily expressed by all users, we also maintain a strong hold that the filtering and merging of profiles should be fully automated. In Section 8.2, we discuss the use of the Resource Description Framework to express both user and device profile. We are currently looking at building a Graphical User Interface that allows the user to easily build his/her profile. The profile representation should make it easy to introduce an automated selection engine. This selection engine will filter the user's profile according to certain rules specified in the profile. Another engine is also needed to merge the user profile with the device profile to construct the "current user profile at this machine". Merging these current profiles of all participants has also to be done before the session starts.
- (d) **Dynamic download and invocation of new services:** One of the most interesting characteristics of the Internet is the ease of service creation. The creation of a service is as easy as creating a web page. Being able to locate needed services and using them on the fly is the subject for the next few sections. It is also possible that a user might ask his Internet Service Provider (ISP) to install the service on a network node, and the user will use this node as a proxy. Users with limited processing power, as well as the ISP, might benefit from this arrangement.

- (e) **Selection of transport requirements:** Given that the user preferences are expressed using high level parameters, a mechanism is needed to translate these preferences into transport requirements (bandwidth, delay, jitter,...). The range of transport options could be wide and selecting the transport parameters that meet the user expectation and increase his/her satisfaction is a difficult problem. Section 9 introduce a novel approach for the selection of QoS parameters.
- (f) **Dynamic request for bandwidth:** The deregulation in the telecommunication industry has created a competitive environment among the bandwidth providers. The user has now the option to choose between various providers. An automated selection of a provider that meets the user or application requirement is achieved through the use of a bandwidth broker.

Locating the device where the user can be currently contacted was the subject of [14]. Dynamic download of services is achieved through the design of an open middleware, and it is discussed in the next two sections. The rest of the paper introduces our approach for filtering and merging user and device profiles. The topic of Bandwidth Broker is not covered here.

#### 4. Middleware Architecture

The introduction of Advanced Intelligent Network (AIN) was considered as a big advance for the old legacy Public Switch Telephone Network (PSTN). The AIN architecture was based on the separation between the switching plane and the control plane. Services became easily created and uploaded into special nodes, the Intelligent Service Control Points (ISCP). The AIN service architecture is well suited for the PSTN network because of the following characteristics:

- It is network centric: all the service logic (intelligence) and processing power resides inside the network.
- The business model assumes that there is only one service provider in each geographical area that provides the communications service.

Even with its advantage, the AIN architecture is not proper for the Internet. The advance in the hardware and software, the deregulation of the market, and the ease of service creation in the Internet call for a new service model where:

- The network only provides connectivity between hosts, unaware of the applications running on the host machines. End devices are powerful enough to host the intelligence and control of the communication session. A suitable service model would be an application centric service model.
- Services are easily created and provided by anyone on the Internet who can afford a server. A desired service model should allow services to be downloaded dynamically and plugged in easily into the user environment. Nowadays, there are plenty of services available in the network that could easily be integrated into the user environment.

Based on the previous observations, we envision that a middleware platform that is flexible and dynamic is a must in order to benefit from the advantages of the Internet. This middleware should hide the vast space of available service on the Internet and provide a simple and easy interface to the user.

Figure 1 depicts our proposed architecture for this middleware. The architecture has two main components: the middleware framework and the services.

The middleware framework is designed with the theme of enabling new services to be added easily to the framework. The framework only defines the interface, and services from different providers can be plugged in as long as they confirm to this interface. This will enable different competing implementation of the same service coming from different service providers. It also opens the network for value-added services, and additional functionalities could be easily introduced into the network. An analogy to this architecture is the Dynamic Link Library (DLL) files for some operating system. These DLL files are called by the operating system or other programs in order to communicate with a specific

device such as a printer or scanner. While the DLL files and the calling programs reside on the same machine, services (or service code) could be located anywhere in the network. The logic for locating, downloading, and instantiating the needed services should be a part of the middleware framework (Section 5)

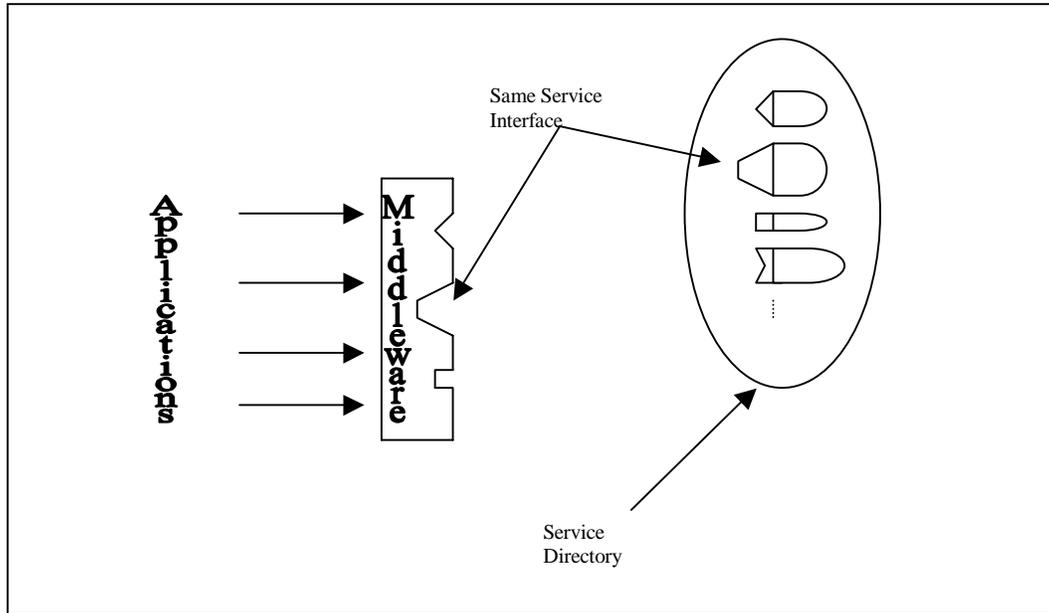


Figure 1. Middleware Architecture

In this architecture, a service appears as an object that provides certain functionalities and has an interface that hides the implementation. Examples of service could be: 1-800 number translation, printing, secret key session generation, media transformer, etc. Services could be provided by different players, and they might differ in their implementation, performance, and price. Service could be either located in the home domain (Jini) or in the Internet (SLP). To make use of any service, the framework has to know the interface of the service. The next section describes how to locate services based on their interface type and possibly, other attributes.

## 5. Locating Services in the Intranet and the Internet

Locating the services available for a user in a certain domain has gained a lot of interest in the last few years. The increase in the number of available services has made the job of system administrator very difficult. To palliate this problem, various architectures and protocols have been proposed to automate the discovery and configuration process. Some of these architectures and protocols are suited for small domains such as corporate or university campus (Jini, Service Location Protocol), while others are more appropriate for the Internet (Lightweight Directory Access Protocol (LDAP), iSLP).

The purpose of the Jini architecture [13] was to assemble a group of hardware and software components into one single distributed system. The resulting system has the properties of flexibility, dynamic configuration and ease of administration. Components can join or leave the confederation seamlessly. The Jini architecture has a look-up service where new services could be registered. Two protocols (Discovery and Join) are used by a service provider to add their services to the lookup service directory. A third protocol, called the "lookup" protocol, it is used when a client or user needs to locate

and invoke a service. Services are identified by their interface type and possibly, other attributes. A similar architecture called the CORBA Trading Service[15] is presented by the Object Management Group (OMG). As a client to the trading service, a service provider can "export a service offer" to the trading service under a defined service type. A client looking for a certain service may "import a service" offer from the trading service.

Another similar architecture from the svrloc working group at the IETF is documented in RFC2165[12]. The proposed architecture, called the Service Location, provides a dynamic configuration for services in a local area network. The proposed protocol, Service Location Protocol (SLP), is used between the User agent and the Directory Agent that offers a centralized database for registered services. A variation of the SLP called Internet SLP (iSLP)[#####] can be used for the Internet.

The Lightweight Directory Access Protocol (LDAP) is software protocol that could also be used to locate resources and services in the intranet or Internet. While LDAP provides access to the X.500 Directory, it does not incur high resource requirements as the Directory Access Protocol (DAP). An LDAP directory is organized in a simple "tree" hierarchy consisting of the different levels: "root" directory, countries, organizations, departments, and individuals. Individuals include services, people, files, and shared resources such as scanners and printers.

All of these mentioned protocols open the space for the end-user to download and initiate services on the fly. The choice between local and Internet wide download depends on the policy set by the system administrator. Service providers are now invited to create services that they think might be useful for their end-users and hence for their business.

## **6. Related Work on User and Device Profile**

Early work in the Multiparty Multimedia Session Control (mmusic) [4] working group at the Internet Engineering Task Force (IETF) resulted in the publication of the Session Description Protocol (SDP) [5] draft. SDP was designed for the Mbone environment, and "is used to advertise multimedia conferences and communicate the conference addresses and conference tool-specific information necessary for participation.". SDP is mostly concerned with the media type and the configuration of communication channels, but it does not reflect the user's preferences and device capabilities.

The CONNEG working group at IETF is currently working on defining a framework [6] for content negotiation between Internet applications and Internet resources. Part of the framework is to describe the range of media features that can be handled and processed by the sender and the receiver. In [7], the format and the registration procedure for individual media features is outlined. The group is also working on the development of a media feature matching algebra based on predicate calculus.

The World-Wide Web Consortium (W3C)[8] has recently begun to define mechanisms for describing and transmitting information (Metadata) about the capabilities and properties of resources on the World Wide Web. The Composite Capabilities/ Preference Profile (CC/PP) describes the way to represent these capabilities/properties using the Resource Description Framework (RDF). The metadata for a document on the Internet could be the author of the document, its language, its content type and its creation date. A profile for a device connected to the Internet may cover the hardware platform (CPU model, the size of the memory, microphone, speaker, camera, ..), system software (operating system, list of audio and video encoders,...) and applications (JMF, Vic, Vat, whiteboard,...) available on the device. Figure 2 shows an example of one device profile.

Hardware	
CPU	AMD Athlon
MemorySizeinMB	128
VoiceInputCapable	Yes
VoiceoutputCapable	Yes
VideoInputCapable	Yes
VideooutputCapable	Yes
Software	
VideoCodecs	MPEG-1 MPEG-2 H.261 H.263
AudioCodecs	MP3 G.711
Applications	JMF Vic Vat WhiteBoard

Figure 2. Device Profile

Work in the Wireless Application Protocol (WAP) Forum [9] is underway to incorporate the CC/PP emerging standard into the wireless network. Digital cellular devices are characterized by less powerful CPUs, less memory (ROM and RAM), restricted power consumption, smaller displays, different input devices (e.g., a phone keypad, voice input, etc.); enabling the interconnection of the Web and wireless terminals requires the wireless device to expose its capabilities to the Web.

In all the previous mentioned works, the concept of user preference was absent. Most of the work concentrated on device configuration and media formats. Even though device capabilities and media formats are important, still one more centric factor is missing from the equation: the end-user.

Early work on QoS user profile can be found in [10]. A user interface is presented that allows the creation of the user profile. The user specifies the *desired* and *maximum* quality for the audio and video media (frame rate, resolution, audio quality). Three Importance values were required for each media type. These preferences are stored in the user profile and used in the *Cooperative QoS Management* framework to select media variants from the news-on-demand server. The work did not discuss how to combine different preferences, and there was no mention of device capabilities.

In [11], the authors presented a framework that uses the user profile data to select the best setting of QoS parameters in the space limited by the boundaries. The user of the model has to select the *minimum acceptable* and *ideal* level for each media type, together with the *sensitivity* of the user to the changes. To represent the user's perceptual quality, the paper introduced the concept of *user satisfaction* for each media type. The framework also presented a model for combining the satisfactions of different media into total satisfaction. The cost criterion was used to select between different parameters with equal satisfaction values. The authors did not consider though the device capabilities, and the selection criterion was based on one-end user only. Section 5 gives a detailed explanation of this work and talks about suggested modifications to the model.

## 7. Video Telephony and Conferencing: Problem and Suggested Solution

The advances in network technology have changed the face of the world. From publishing, education, training, medicine, to entertainment, there is rarely a field in life that is not affected by these advances. Many new services, especially multimedia services are becoming indispensable parts of our daily life.

Advances in network technology did not only provide users with a multi-service multimedia platform, but it also enhanced service mobility with new technologies such as cellular telephony and mobile IP, to name a few. In addition to device mobility, personal mobility is becoming now the focus of many research projects. In the near future, users are expected to roam across different networks, using the services provided in the domain they are currently visiting. Users will also be able to use devices and services available in a visiting domain, provided that there exists an exchange policy for authentication and charging between the visiting domain and the user's home domain.

With these advances, new problems come to surface. The heterogeneity of devices, ranging from small devices to personal computers, coupled with the variety of media types, has always been an issue for multimedia communication. This diversity in communication devices has always posed limitation on compatibility. Users with non-compatible software sometimes receive media that they can not store or playback properly. Users having access to different devices should also have different configuration files for each device. This puts extra burden on the user, and encumbers personal mobility.

To benefit from all these advances and limit the drawbacks of heterogeneity, we suggest that each device has a profile that stores information about the hardware and software on that device. We also suggest that each user keeps his/her personal profile on a profile server in his home domain. A profile manager is required to provide users with the ability to create and modify their profiles. Profile managers should also provide a communication interface to profile managers in other domains to support personal mobility. The next sections discuss the content and use of these profiles.

## **8. User Profile and its Possible Representation**

### **8.1. User Profile**

In plain computing environment, it is only sufficient to keep an encrypted version of the user password in his/her profile to give him/her access to certain computing facilities. When a user types in his/her user id and his/her password, the system usually contacts an authentication server to determine the authenticity and authority of the user. In a multi-user Unix environment, an application may also be configured using environment parameters or using the “*dot rc*” file (.pinerc, .newsrsrc,...). Certain applications under WindowsNT such as Netscape also support multi-user configuration using the user profile. Even though the parameters in the “*dot rc*” or user profile are specified by the user, they were specific to the application.

In this paper, we are suggesting the use of a user-centric profile. This profile should capture the personal properties and preferences of the user. Properties include, but are not restricted to full name, employer, email account, phone number, class factor (described later) etc.. As for preferences, they can cover the parameters for all applications to which the user has access to, but we are interested here in the user's preferences concerning the quality of multimedia communication. These preferences cover the choice of receiving/sending audio and video such as the frame rate, the resolution, the audio quality and the degree of preference (weight) for every media types. Certain preferences are also related to the selection quality of individual media types for communication with particular persons. A public section for preferences may be used to specify that the user is interested in news about the stock market, and if so, the prices for the shares of certain companies. This section, we call it *public preferences section*, can be used by content providers to provide tailored advertisements and services. Figure 3 shows an example of a user profile.

<p align="center"><b>Personal Identification</b></p> Name: Peter Alleyene Employer: University of Ottawa Email: <a href="mailto:peter@site.uottawa.ca">peter@site.uottawa.ca</a> Phone Number: 613-562-5800 ClassFactor: 0.5		<p align="center"><b>Callees</b></p> Callee: <a href="mailto:rami@wam.umd.edu">rami@wam.umd.edu</a> QoSselectionWeight: 0.6 PriceCeiling_CentsPerMinute: 15 AudioWeightFactor: 9 VideoWeightFactor: 2	
<p align="center"><b>Public Preferences</b></p> Music: Jazz Blues Sport: Soccer Tennis		Callee: <a href="mailto:john@nortelnetworks.com">john@nortelnetworks.com</a> QoSselectionWeight: 0.75 PriceCeiling_CentsPerMinute: 30 AudioWeightFactor: 1 VideoWeightFactor: 1  Callee: <a href="http://www.sportsnews.com">www.sportsnews.com</a> QoSselectionWeight: 0 PriceCeiling_CentsPerMinute: 18 AudioWeightFactor: 0 VideoWeightFactor: 9  Callee: <a href="http://www.newsondemand.com">www.newsondemand.com</a> QoSselectionWeight: 0 PriceCeiling_CentsPerMinute: 11 AudioWeightFactor: 7 VideoWeightFactor: 0	
<p align="center"><b>VideoPreferences</b></p> ReceiveVideo: Yes SendVideo: Yes FrameRate: Min Acceptable 10 Ideal 30  FrameResolution: Min Acceptable 320x240 Ideal 800x600 Sensitivity Parameter 6		<p align="center"><b>AudioPreferences</b></p> ReceiveAudio: Yes SendAudio: Yes Min Acceptable: Telephone Quality Ideal: CD Quality	

Figure 3. User Profile

The profile in Figure 3 shows the profile for a user named *Peter Alleyene*. The first part of the profile represents the personal information of the user. The *Video Preferences* section describes the user's preferences to send and receive video. It also shows what is the *Minimum Acceptable* and *Ideal* values for the frame rate and resolution of the video. The *Sensitivity Parameter*, explained in Section 5, reflects the sensitivity of the user to the changes in the parameter value between the minimum and maximum rate. The same holds for the *Audio Preferences* section. In the *Callees* section, each *callee* has a unique user id (email or URL address) and a *QoSselectionWeight* parameter. This parameter will help decide how much the preferences of the callee influence the selection of the session parameters (see Section 5.2). The other two parameters, *AudioWeightFactor* and *VideoWeightFactor* reflects the preference of the user to use respectively audio and video to communicate with this callee. The *PriceCeiling\_CentsPerMinute* tells the system what is the ceiling on the price if the user wants to communicate with this callee.

## 8.2 Representation of the User and Device Profile using the CC/PP Protocol

To represent the properties and preferences of the user in the user profile and the device profile, we consider the use of the CC/PP protocol. Using the RDF format will eliminate the interoperability problem for meta-data representation at various levels (user, hardware and software component). The W3C, IETF, and WAP Forum are working closely to avoid unnecessary divergence between the recommendations and standards. We strongly believe that using a common unified technology in the metadata representation will help resolve the interoperability problem between different levels.

Another reason for using the RDF model for representing user profiles is the automation in the selection of media types and parameters. Automating the selection procedure requires the preferences to be represented in a machine-understandable format. Figures 4 and 5 shows the previous user and device profile expressed in the CC/PP format. Details about the CC/PP are available from the W3C Forum.

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:prf="http://www.site.uottawa.ca/school/research/DSRLab/elkhatib/QoS_User_Schema.0#">
<rdf:Description ID="MyProfile">
<prf:Defaults rdf:resource="http://www.site.uottawa.ca/profiles/gradProfile/">
</overwrite the default properties .....->

<prf:Component>
<rdf:Description ID="PersonalIdentification ">
<rdf:type resource=" http://www.site.uottawa.ca/school/research/DSRLab/elkhatib/QoS_User_Schema.0/#PersonalIdentification
"/>
    <prf:Name>Peter Alleyne</prf:Name>
    <prf:employer>University of Ottawa</prf:employer>
    <prf:Email>peter@site.uottawa.ca</prf:Email>
    <prf:PhoneNumber> 613-562-5800 </prf:PhoneNumber>
</rdf:Description>
</prf:Component>

<prf:Component>
<rdf:Description ID=" VideoPreferences ">
<rdf:type resource=" http://www.site.uottawa.ca/school/research/DSRLab/elkhatib/QoS_User_Schema.0/# VideoPreferences" />
    <prf:SendVideo>Yes</prf:SendVideo >
    <prf:ReceiveVideo>Yes</prf: ReceiveVideo >

    <prf:MinAcceptableFrameRate> 10 </prf:MinAcceptableFrameRate>
    <prf:MaxAcceptableFrameRate> 30 </prf:MaxAcceptableFrameRate>
    <prf:FrameRateSensitivityParameter> 8 </prf:FrameRateSensitivityParameter>

    <prf:MinAcceptableFrameResolution > 320x240 </prf:MinAcceptableFrameResolution >
    <prf:MaxAcceptableFrameResolution > 800x600 </prf:MaxAcceptableFrameResolution >
    <prf:FrameResolutionSensitivityParameter> 6 </prf:FrameResolutionSensitivityParameter>

</rdf:Description>
</prf:Component>

<prf:Component>
<rdf:Description ID=" AudioPreferences ">
<rdf:type resource=" http://www.site.uottawa.ca/school/research/DSRLab/elkhatib/QoS_User_Schema.0/#AudioPreferences" />
    <prf:SendAudio>Yes</prf:SendAudio >
    <prf:ReceiveAudio>Yes</prf: ReceiveAudio >
    <prf:MinAcceptableAudioQuality> Telephone </prf: MinAcceptableAudioQuality >
    <prf:MaxAcceptableAudioQuality> CD </prf:MaxAcceptableAudioQuality>
    <prf: AudioQualitySensitivityParameter> 5 </prf: AudioQualitySensitivityParameter>

</rdf:Description>
</prf:Component>

<prf:Component>
<rdf:Description ID="Public_Preferences">
<rdf:type resource="http://www.site.uottawa.ca/school/research/DSRLab/elkhatib/QoS_User_Schema.0/#Public_Preferences" />
    <prf:Music>
        <rdf:bag>
            <rdf:I> Jazz</rdf:I>
            <rdf:I> Blues </rdf:I>
        </rdf:bag>
    </prf:Music>

    <prf:Sport>
        <rdf:bag>
            <rdf:I> Soccer </rdf:I>
            <rdf:I> Tennis </rdf:I>
        </rdf:bag>
    </ prf:Public_Preferences>
</rdf:Description>
</prf:Component>

</rdf:Description>
</rdf:RDF>

<prf:Component>
<rdf:Description ID=" CalleeList ">
<rdf:type resource=" http://www.site.uottawa.ca/school/research/DSRLab/elkhatib/QoS_User_Schema.0/#CalleeList" />

    <prf:Component>
    <rdf:Description ID=" Callee ">
    <rdf:type resource=" http://www.site.uottawa.ca/school/research/DSRLab/elkhatib/QoS_User_Schema.0/#Callee"
/>
        <prf:CalleeId>rami@wam.umd.edu </prf:CalleeId>
        <prf:ClassFactor> 0 </prf:ClassFactor>
        <prf:AudioWeightFactor> 9 </prf:AudioWeightFactor>
        <prf:VideoWeightFactor> 2 </prf:VideoWeightFactor>
    </rdf:Description>

```

```

</prf:Component>
:
:
</rdf:Description>
</prf:Component>

```

Figure 4. User Profile of Figure 3 Expressed in RDF

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:prf=" http://www.site.uottawa.ca/school/research/DSRLab/elkhatib/deviceschema1.0#">
<rdf:Description ID="MyDevice">

<prf:Component>
<rdf:Description ID="TerminalHardware">
<rdf:type resource="http://www.wapforum.org/UAPROF/ccppschem1.0/#HardwarePlatform" />
  <prf:Defaults rdf:resource="http://www.DeltaComputers.com/profiles/PC2160" />
  <!-- override the ImageCapable property and add VoiceInputCapable and Keyboard properties -->
  <prf:CPU>AMD Athlon </prf:CPU>
  <prf:MemorySize> 128 </prf:MemorySize>
  <prf:VoiceInputCapable>Yes</prf:VoiceInputCapable>
  <prf:VoiceoutputCapable>Yes</prf:VoiceoutputCapable>
  <prf:VideoInputCapable>Yes</prf:VideoInputCapable>
  <prf:VideooutputCapable>Yes</prf:VideooutputCapable>

  </rdf:Description>
</prf:Component>

<prf:Component>
<rdf:Description ID="TerminalSoftware">
<rdf:type resource="http://www.wapforum.org/UAPROF/ccppschem1.0/#SoftwarePlatform" />
<prf:Defaults rdf:resource="http://www.symbian.com/profiles/pda/epoc" />
  <!--Specify VideoInputEncoder and AudioInputEncoder properties -->
  <prf:VideoCodecs>
    <rdf:Bag>
      <rdf:l>MPEG-1</rdf:l>
      <rdf:l>MPEG-2</rdf:l>
      <rdf:l>H.261</rdf:l>
      <rdf:l>H.263</rdf:l>
    </rdf:Bag>
  </prf:VideoCodecs>

  <prf:AudioCodecs>
    <rdf:Bag>
      <rdf:li>MP3 </rdf:li>
      <rdf:li> G.711</rdf:li>
    </rdf:Bag>
  </prf:AudioCodecs>

</rdf:Description>
</prf:Component>

</rdf:Description>
</rdf:RDF>

```

Figure 5. Device Profile of Figure 2 Expressed in RDF

## 9. Automatic QoS Parameter Selection

Before engaging in a telecommunication session, a decision has to be made as to what hardware components to use (camera, speakers, microphone,...), which software to start (Vic, Vat, whiteboard), what type of media to use (MPEG1, H.261), and the quality of the presentation of the media (frame rate, frame resolution, audio quality,...). This decision is affected by many factors including the hardware and software profile of the devices to be used, referred to as limitations, and the properties, preferences, and privileges of end users. This suggests that the device and user profiles of all end-users should be merged to come up with a profile for the communication session. We consider in the following a communication session between two users.

The first step in the selection procedure consists of selecting the possible interoperable hardware and software components in the devices of the two users. A device that does not have an MPEG-1 codec renders the communication with other devices sending MPEG-1 media type impossible. Also a device which does not have a speaker can not support a telephony session. To construct the list of compatible components, the profile of all devices should be merged together. Information about compatibility could be stored in a central database, and fed into the selection process.

As an example, let us assume that one of the devices is described by the profile of Figure 2, and that this second device has the following components: camera, speaker, MPEG-1 codec, G.711, MP3 codec. Merging the two profiles of the devices should reveal that MPEG-1 and G.711 codecs could be used. While the two devices can send and receive video media, one of the devices can only receive audio media since it does not have a microphone.

The second step in the selection procedure consists of selecting specific parameter values for certain QoS parameters, such as video frame rate and resolution, within the bounds imposed by the limitations of the device hardware and available network service. For instance, the network access line of one of the devices may limit the maximum throughput (e.g. modem or wireless connection). Another device may have a small display area which limits the resolution of the video. Another limitation may be imposed by the user by establishing a maximum cost per minute (which may depend on the effective network throughput used). Within these limits, the most appropriate QoS parameter values should be selected during this second step. In order to optimize this selection process, it is important to consider the user's preferences, as indicated in his user profile.

### 9.1. Quantifying the user preferences

In order to take the user preferences into account during this second step of QoS parameter selection, we have to quantify the preferences of the user. An interesting approach has been described in [11]. The paper considers several QoS parameters  $x_i$  representing different aspects of the quality of various multimedia components and considers the user's appreciation of these parameters. The appreciation of a user is expressed by a "satisfaction function"  $g_i$  which determines the "satisfaction"  $s_i$  of the as a function of the value  $x_i$  of the corresponding parameter, that is,  $s_i = g_i(x_i)$ . The values of the satisfaction lie between the values zero and one (as indicated in Figure 6), which corresponds to the minimum acceptable value (M) and an ideal value (I).

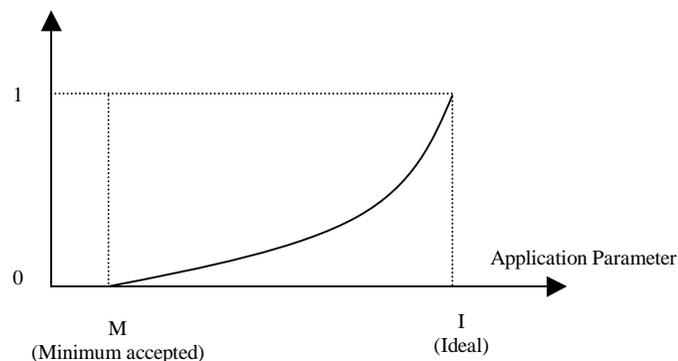


Figure 6. Example of the mapping function determining the user satisfaction depending on the value of a QoS

It is noted the satisfaction function  $s(x)$  should increase monotonically. Since the exact form of the user satisfaction function is difficult to determine, various forms of monotonically increasing functions are proposed in [11]. In this paper we assume the simplest case where the satisfaction function is linear between the two points of minimum acceptable and ideal values, and constant outside that interval. This means that  $s(x)$  is zero for values of  $x$  smaller than  $M$ , equal to one for value of  $x$  larger than  $I$ , and for values in between  $M$  and  $I$  the satisfaction is given by the formula

$$s(x) = \frac{1}{I - M}(x - M)$$

In the case when there are more than one application parameter their respective satisfaction functions, a combined satisfaction  $S_{tot}$  should be computed. The authors of [11] propose combination function  $f_{comb}$  which determines the total satisfaction  $S_{tot}$  from the satisfactions  $s_i$  for the individual parameters as follows:

$$S_{tot} = f(s_1, s_2, s_3, \dots, s_n) = \frac{n}{\sum_{i=1}^n \frac{1}{s_i}} \quad (\text{Equa. 1})$$

$f_{comb}$  has the following properties:

- 1- One individual low satisfaction is enough to bring the total satisfaction to a low value. This property of the function is essentially saying that if the user is not satisfied with one of the parameters, then he is not satisfied in total.
- 2- The total satisfaction of equal individual satisfactions  $s_i$  is equal to the satisfactions  $s_i$ .

We note that  $f_{comb}$  is a many to one mapping function, and hence different combinations of individual satisfaction values are possible for one value of  $S_{tot}$ . To find out what the best possible combination of individual satisfaction functions is, another selection criterion is needed. The most reasonable selection criterion is the cost. Providing a tariff structure which determines the cost for the different values  $x_i$  of the individual application parameters, one can devise an optimization strategie for finding application parameter values that minimize the cost for a given global  $S_{tot}$  [11].

## 9.2 Weighted Combination of User's Preferences

We think that the approach described above is a major step towards a simple user-friendly interface for user level QoS specification, however, further considerations could be taken into account as described below. A first improvement results from the observation that users in telecommunication session might find some media types more important than others. For instance, a user of a news-on-demand service might prefer to receive high quality audio with low quality video as compared to average quality audio and average quality video. In the case of a user watching a sport event the situation may be the opposite (if the user does not care about the audio of the commenter).

This preference to individual media can play a factor when it comes to the calculation of the total satisfaction  $S_{tot}$ . By assigning different weights  $w_i$  to the different parameters  $x_i$ ,  $S_{tot}$  will reflect the user

preference for different media types. The combination function for the total user satisfaction can be redefined as follows:

$$S_{tot}^{user} = f_{comb}(s_1, s_2, s_3 \dots, s_n, w_1, w_2, w_3 \dots, w_n) = \frac{n\bar{w}}{\sum_{i=1}^n \frac{w_i}{s_i}} \quad (\text{Equa. 2})$$

where  $w_i$  is the weight for the individual satisfaction  $s_i$  and  $\bar{w} = \frac{\sum_{i=1}^n w_i}{n}$ .

These constants weights factors are selected by the user, and they are stored in the user profile (AudioWeightFactor, VideoWeightFactor,..). The selection of these weights depends on the type of service the user is willing to receive when using a specific service or communicating with a given callee.

### 9.3. Combining Preferences of Several Users

So far we have only considered the QoS preferences of a single user. But all conversational multimedia applications involve several users. It is therefore important to determine how the possibly conflicting preferences of the different users are reconciled in order to come up with QoS parameters which are suitable for all participating users.

In certain circumstances, some given parameters may be determined simply based on the preferences of a single user. This may be the case in a two-way teleconference between two users A and B, where the parameters of the video visible by User A would be determined based on the preferences of User A alone, and the video in the opposite direction based on the preferences of User B. However, the situation may be more complex if the cost of the communication is paid by User A and the selection of the video received by User B has an impact on the communication cost.

In other circumstances, as for instance in the case of the joint viewing of a video clip by several participants in a teleconference, the selected quality parameters should be determined based on the preferences of all participating users. In such circumstances, we propose to use the same combination function for user satisfaction considered above and (optionally) introduce a weight for each of the participating users, called the *QoS selection weight*, which determines how much the preferences of the user influence to overall QoS parameter selection. The total satisfaction (computed for all users) is then given by

$$S_{tot} = f_{comb}(s_{tot}^{usr_1}, s_{tot}^{usr_2} \dots, s_{tot}^{usr_n}, a_1, a_2 \dots, a_n) \quad (\text{Equa. 3})$$

where  $s_{tot}^{usr_i}$  is the total satisfaction for user  $i$ , and  $a_i$  is the *QoS selection weight* for user  $i$ . In the case that the weight of a given user is zero, the preferences of this user are not taken into account for the selection of the QoS parameters.

## 10. Example Scenario (Continue)

Let us assume that during the communication session, Alice and Bob decided to view a video clip. We assume that the hardware and software constraints on the terminal A and B allow for the following values of QoS parameters: (a) video frame: 10, 12, 14, ...30 frames per second, and (b) resolution: 180x240, 240x320, 300x400, 360x480, 420x560, or 480x640. For each of the possible combination of frame rate and resolution values, we may determine (using Equation 2) the satisfaction of each of Alice and Bob based on their respective QoS preferences stored in their user profile. Some of these combination may not be feasible because the corresponding throughput can not be supported by the network access interface of one of them.

We may now use Equation 3 to determine the overall user satisfaction, taking into account the QoS selection weights of both Alice and Bob. These weights may be determined by Alice since she initiated the teleconference and she is paying for its cost. Examples of such weights are indicated in the user profile of Figure 3. If the communication cost is no issue, the selection of the QoS parameters would

be based on the optimization of the overall user satisfaction, that is, the combination of frame rate and resolution that optimizes the overall user satisfaction will be adopted.

This means that if the weight of Bob is zero, only the preferences of Alice will be used for the selection of the video QoS parameters. If Alice has specified in her profile a high weight for frame rate and a low weight for resolution, the selected video quality will trade-off high frame rate against low resolution when the available network throughput limits be available possibilities. On the other hand, if the QoS selection weights of Bob is larger than for Alice and the user profile of Bob specifies low weight for frame rate and high weight for resolution, then the optimization will follow more the preferences of Bob, that is, a video quality with higher resolution will be selected.

## **11. Conclusion**

A middleware architecture that insulate the user from network is presented. This architecture enables easy introduction and use of new services into the Internet. To control and select the QoS parameters for a communication session, a user profile that keeps the user's preferences for each media type is suggested. The combination of the user preferences and the limitation of the used device is outlined. We also presented an extension to the framework presented in [11] to enable weight assignment to different preferences. The extended framework will also considers the preferences of the different participants in a session.

## References

- [1] The ATM Forum Technical Committee. *Traffic Management Specification Version 4.0, af-tm-0056.000*, Apr. 1996
- [2] P.P. White and J. Crowcroft. The integrated services in the internet: State of the art. Proceedings of the IEEE, 1997.
- [3] IETF, "An architecture for differentiated services," RFC 2475, Dec. 1998.
- [4] IETF, <http://www.ietf.org/html.charters/mmusic-charter.html>
- [5] IETF, "SDP: Session Description Protocol," RFC 2327, Apr. 1998.
- [6] IETF, "Requirements for protocol-independent content negotiation," RFC2703, Sep. 1999.
- [7] IETF, "Media Feature Tag Registration Procedure," RFC 2506, March 1999.
- [8] W3C <http://www.w3.org/>
- [9] WAP, <http://www.wapforum.org/>
- [10] S. Fischer, A. Hafid, G.v. Bochmann and H. de Meer. "Cooperative quality of service management for multimedia applications". Proceedings of the 4th IEEE International Conference on Multimedia Computing and Systems, Ottawa, Canada, June 1997, pp. 303-310.
- [11] A. Richards, G. Rogers, V. Witana, and M. Antoniadis. Mapping User Level QoS from a Single Parameter. In Second IFIP/IEEE International Conference on Management of Multimedia Networks and Services, Versailles, November 1998.
- [12] IETF, "Service Location Protocol," RFC 2165, June 1997.
- [13] Sun Microsystems, <http://www.sun.com>
- [14] X. He, K. El-Khatib, G. v. Bochmann. "QoS Enabled Middleware Based on Home Directory Agent". To be published.
- [15] Object Management Group. OMG RPF5 Submission: Trading Object Service, May 1996