A Quality of Service Negotiation Procedure for Distributed Multimedia Presentational Applications

Abdelhakim Hafid[1], Gregor v. Bochmann[1] and Brigitte Kerhervé[2]

[1]Université de Montréal, Dept. d'IRO, Groupe de Téléinformatique,
Montréal, H3C 3J7, Canada
[2]Université du Québéc à Montréal, Dpt. d'Informatique,
CP 8888, Montréal, H3C 3P8, Canada

Abstract: *Most of current approaches in designing and implementing distributed multimedia (MM) presentational applications, e.g. news-on-demand, have concentrated on the performance of the continuous media file servers in terms of seek time overhead, and real-time disk scheduling; particularly the QoS negotiation mechanisms they provide are used in a rather static manner, that is, these mechanisms are restricted to the evaluation of the capacity of certain system components, e.g. file server, a priori known to support a specific quality of service (QoS). In contrast to those approaches, we propose a general QoS negotiation framework that supports the dynamic choice of a configuration of system components to support the QoS requirements of the user of a specific application: we consider different possible system configurations and select an optimal one to provide the appropriate QoS support. In this paper we document the design and implementation of a QoS negotiation procedure for distributed MM presentational applications, such as news-on-demand. The negotiation procedure described here is an instantiation of the general framework for QoS negotiation which was developed earlier. Our proposal differs in many respect with the negotiation functions provided by existing approaches: (1) the negotiation process uses an optimization approach to find a configuration of system components which supports the user requirements, (2) the negotiation process supports the negotiation of a MM document and not only a single monomedia object, (3) the QoS negotiation takes into account the cost to the user, (4) the negotiation process may be used to support automatic adaptation to react to QoS degradations, without intervention by the user/application.*

Keywords: *QoS management, news-on-demand, negotiation, adaptation*

1. Introduction

Multimedia information systems integrate diverse media, such as text, video and images, to enable a range of MM applications such as news-on-demand [Vel 95]. Multimedia news-on-demand is the target application for the CITR major project "Broadband Services". News-on-demand prototype is an integration of software components developed by the various teams working on different components of the major project. The prototype consists of the distributed multimedia database (DBMS) from University of Alberta [Vit 95], the distributed continuous media file (CMFS) server from University of British Columbia [Neu 96], the synchronization component from University of Ottawa [Lam 94], a scalable video decoder from INRS telecommunications [Dub 95], and the QoS management component from University of Montreal [Haf 95c].

In the context of our sub-project "QoS Negotiation and adaptation" we made a set of proposals concerning QoS management for distributed MM applications [Haf 95a, Haf 95b, Haf 95c, Haf 95d, Haf 96]. Particularly, we developed a general framework for QoS management [Haf 95c] based on the existing QoS architectures. In contrast to those architecture [Cam 94, Fer 95, Vol 95, Gop 94, Zha 95, Kes 95, Nah 95] we proposed a QoS management architecture that supports the dynamic choice of a configuration of system components to support the QoS requirements of the user of a specific application: we consider different possible system configurations and select an optimal one to provide the appropriate QoS support. Our proposals have been described at an abstract level without any assumptions with respect to the type of the applications considered, and hardware and software infrastructure in use.

In this paper we use this general framework for QoS negotiation to design and implement a negotiation proce-

1082-8907/96 $5.00 © 1996 IEEE
Proceedings of HPDC-5 '96

dure for MM presentational applications, such as news-on-demand. The here described negotiation procedure assumes a system which consists of client machines, a set of server machines, and networks connecting these machines. The objective of the QoS negotiation procedure is to find an optimal system configuration which might support the delivery of a MM document requested by the user, at a given client machine, with the desired QoS. The user selects the document he/she wants to play and selects a QoS user profile. The QoS negotiation procedure uses this information (1) to find the system components, namely the server machine and the network, which might support the user requirements, or (2) to reject the user request. The goal is not only to find components which have enough resources to support the user request, but also to maximize the resource utilization and/or to minimize the cost the user will be charged. If two component configurations have enough resources to support the requested service, the negotiation procedure should be able to select the optimal configuration, that is, optimization parameters should be defined. Furthermore the best QoS that can currently be supported must be returned to the user, when his/her request cannot be supported in its original form.

The QoS negotiation procedure we propose has the following characteristics:
(1) It uses an optimization approach to find a configuration of system components which supports the user requirements.
(2) It supports QoS negotiation for a MM document, and not only for a single monomedia object.
(3) It takes into account the cost constraints.
(4) It supports automatic adaptation to react to QoS degradations without the direct intervention by the user/application.

The paper is organized as follows. Section 2 describes the model used to specify a MM document. Section 3 presents the user profile notion which enables the (human) user to specify his/her requirements. The main steps of the negotiation procedure are described in Section 4. Section 5 defines the optimization scheme used during the negotiation phase. Section 6 describes the function used to map the user requirements into network QoS parameters. Section 7 presents the scheme used to compute the cost the user must pay to get the requested service. Section 8 presents the graphical user interface (GUI), for QoS negotiation which we implemented. Finally, Section 8 concludes the paper.

2. Multimedia document

A MM document could be multimedia or monomedia. A multimedia document is composed of several monomedia. Figure 1 shows the structure of a MM document using the notation of OMT object model [Ram 91]. It shows that a document is either a monomedia or a multimedia, and that a multimedia is composed of one or more monomedia (aggregation links), and has attributes which consists of spatial and temporal synchronization constraints.

A monomedia is defined in a particular medium: a text, a still image, an audio sequence, a graphic or a video sequence. Several physical representations, called variants, may exist for a monomedia object which correspond to a format variant. Each variant has different degrees of qualities. More generally, variants of the same monomedia may differ in terms of some static parameters which concern mainly the format of the coding, the size of the file, the QoS parameters associated with the file, e.g. video color and audio quality, and the localization of the file. For example, two variants of the same video sequence could offer different color qualities; Variant1 could be a super-color variant of the video sequence, while Variant 2 could be the black and white variant of the same video. Copies of the same file are considered also as variants. More detailed description of the model of MM document used can be found in [Boc 96, Ker 96].

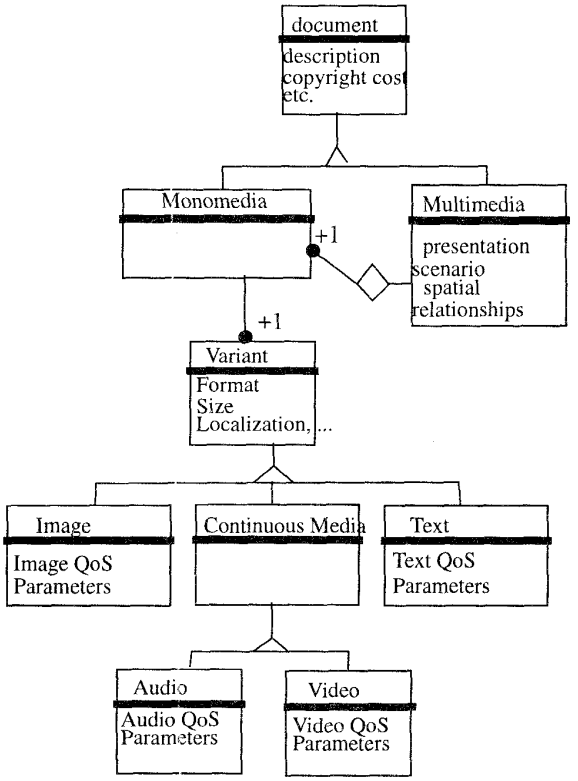

Figure 1. MM document model

3. QoS specification by the user

The user specifies his/her requirements via a graphical user interface, which we call here the QoS GUI (see Section 8). The QoS GUI should hide, as much as possible, the

internal QoS parameters, e.g. throughput and jitter, and provide facilities to describe the requested QoS in terms of a set of user-perceived characteristics of the performance of a service. In addition to QoS and cost requirements, the user should be able to specify some optimization criteria in terms of weights, which we will call importance factors. These factors indicate the degree of importance of different QoS characteristics and the cost. Examples of the utilization of the importance factors are the following: (1) the user specifies that the QoS is more important than the cost, (2) the user specifies that the audio is more important than the video, (3) the user specifies that video frame rate is more important than video resolution, and (4) the user specifies that french is more important than english.

To facilitate the QoS specification by the user, we introduce the notion of user profiles. A user profile describes user preferences in terms of (1) QoS setting for video, audio, still images and text, (2) the cost he/she is willing to pay to play the requested document with the desired quality, (3) time constraints, such as the delivery time, and (4) the importance factors. To decrease the system blocking probability, the QoS GUI provides facilities to set not only the *desired QoS* but also the *worst acceptable QoS* values.

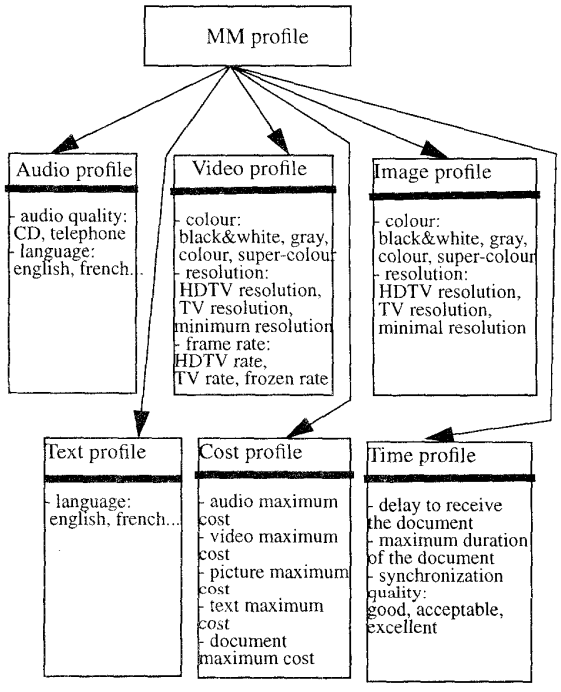

Figure 2. MM profile

A user profile consists of (1) a MM profile which indicates the desired values, (2) a MM profile which indicates the worst acceptable values, and (3) the importance profile which indicates the values of importance factors. A MM profile consists of video, audio, text, and image profiles, cost profile and time profile (Figure 2). The user is able to specify (1) any integer values between HDTV rate (60 frames/s) and frozen rate (1 frame/s) for the frame rate, and (2) any integer values between HDTV resolution (1920 pixels/line) and minimal resolution (10 pixels/line) for resolution. The cost profile attributes should be specified in terms of $, while the time profile attributes should be specified in terms of seconds.

The system component responsible for user profile management via the QoS GUI is called the profile manager.

4. The operation of the QoS manager

The component which implements the QoS management functions, namely QoS negotiation and adaptation, is called the QoS manager. The role of the QoS manager is to determine an optimal system configuration (server(s), network(s), and the client machine) which might support the user requirements. In the context of the news-on-demand prototype, this activity consists of determining, for each monomedia of the document, which of the available variants is the "best". Indeed a variant is stored on a server machine and can be delivered to the client via a network; then given a variant, we are able to determine the corresponding configuration of system components involved in its delivery.

The QoS manager considers possible system configurations, which we will call system offers (see Definition 1), selects an optimal one, and makes an offer, which we call user offer (see Definition 2), to the user. It is worth noting that a user offer can be derived from a system offer using appropriate mapping functions.

Definition 1: A system offer consists of a set of variants (a variant for each monomedia component of the document) and the cost the user should pay.

Definition 2: A user offer represents the QoS the system is able to provide and the cost the user should pay for the document delivery. A user offer is specified as a MM profile (see Section 3).

The input of the negotiation procedure consists of the document to be played and the user profile selected by the user, while the output consists of the negotiation results. The negotiation results consists of the *negotiation status* and possibly a user offer.

The negotiation status can assume five values:

(1) *SUCCEEDED:* indicates that the negotiation phase succeeded, that is, the requested QoS and the maximum cost the user is willing to pay are satisfied by the system. A user offer (which does not violate the worst acceptable values contained in the user profile) is returned.

(2) *FAILEDWITHOFFER:* indicates that the negotiation phase failed, but a user offer which does not satisfy the user requirements but can be supported by the system, is

returned.
(3) *FAILEDTRYLATER:* indicates that the negotiation phase failed because of resources shortage. However, the user may try later and his/her request may eventually be accepted.
(4) *FAILEDWITHOUTOFFER:* indicates that the negotiation phase failed because no possible instantiation of the functional configuration (of the news-on-demand service) to a physical configuration exists, e.g. the client machine does not support a suitable decoder to decode the requested data.
(5) *FAILEDWITHLOCALOFFER:* indicates that the client machine does not support the QoS requested by the user, e.g. the user asks for a color video, while the client machine screen is black&white.

The negotiation procedure consists of five main steps.
(1) Static local negotiation:

This steps allows to check whether the client machine characteristics, such as the screen size and the screen color, support the requested QoS. If the client machine does not support the QoS requested by the user, a notification is sent to the user (*FAILEDWITHLOCALOFFER*), via the profile manager.
(2) Static compatibility checking:

This step allows to check the format compatibility of the variants, related to the document selected, with the decoder(s) supported by the client machine. For example, if the client machine supports only MPEG decoder and the video variant, *variant1,* is coded as MJPEG file then *variant1* will simply not be considered as a feasible system offer. If such an activity produces no feasible system offer, a notification (*FAILEDWITHOUTOFFER*) is sent to the user, via the profile manager.
(3) *Computation of classification parameters:*

For each feasible system offer, two parameters are computed: static negotiation parameter and importance (see Section 3.2). Such parameters are required to sort the system offers from the best to the worst system offer, that is, those parameters are an input to the next step.
(4) *Classification of system offers*:

Given the requested QoS/cost (user profile) and the available audio and video variants of the document, Step 4 allows to sort the different system offers (Video, Audio, image, text, audio&video) from the best system offer (which corresponds to an optimal configuration) to the worst system offer and thus produces an ordered list of system offers. We decided to consider all the feasible system offers even those which do not satisfy the requested QoS and/or cost. Such a decision is motivated by two reasons: (1) if the requested QoS/cost cannot be supported by the system, the best system offer, which does not satisfy the user requirements, is produced, and (2) during the active phase, if QoS violations occur the adaptation procedure makes use of the whole set of feasible system offers to perform an automatic adaptation [Haf 95d].
(5) *Resources commitments*:

The QoS manager considers the best system offer that satisfies the QoS/cost requested by the user, and asks the transport system and the media file servers to reserve resources to support the QoS associated with the system offer. If such an activity is a success, thus the negotiation completes successfully and a notification (*SUCCEEDED*) is sent to the user; otherwise, the next acceptable system offer is considered. If there are not enough resources to support any of the acceptable system offers, the same procedure is applied on the feasible (not acceptable) system offers. If the resources required to support one of those system offers are reserved the negotiation phase is over and a notification (*FAILEDWITHOFFER*) is sent to the user, via the profile manager.
If the whole set of the feasible system offers are considered and no resources are available, a notification (*FAILEDTRYLATER*) is sent to the user, via the profile manager.
(6) *User confirmation:*

Once the resources are reserved for a system offer (Step 5), a notification is sent to the user (the user offer derived from the system offer, is displayed). The user must confirm the user offer (rejection or acceptance) within a limited amount of time since the resources are reserved. If the confirmation is an acceptance, the system starts playing the document; otherwise the resources reserved for the system offer are de-allocated.

During the playout of the document, if the network or/ and the server machine become congested thus leading to lower presentation quality, the QoS manager makes use of the adaptation procedure. In this case, the QoS manager considers the ordered set of system offers, except the current one (which is in difficulty), and executes Step 5. If an alternate system offer is selected and the required resources are reserved, the QoS manager automatically performs a transition from the current system offer to the new one. That is, the delivery of the document will continue using the services of the alternate components (which support the new system offer). In the current implementation, to perform this transition, the QoS Manager stops the presentation of the document after having obtained the current position of the document, and restarts the presentation (using the alternate components) from the position parameter determined earlier. This transition procedure is a simple one, more sophisticated procedures may be used.

5. The classification of system offers

To classify a set of system offers in response to a user request is not an obvious task. To the best of our knowledge there is no literature published which deals with the issue. To classify system offers in terms of cost is obvious, since the cheapest system offer is the best system offer. To

classify system offers in terms of QoS, we proposed in [Haf 96] to use the notion of weighted average: for each system offer we compute the weighted average, *WA* of the QoS parameter values associated with the system offer, and the system offer with the highest *WA* is the best system offer. In the following we use the terms system offer and user offer interchangeably. For sake of clarity we present the examples in terms of user offers instead of system offers; a single system offer corresponds to a single user offer.

The classification of the offers in terms of only QoS or only cost, is neither optimal nor suitable to perform "smart" negotiation. If we classify the offers in terms of cost (resp. QoS), it is likely that the "best" offer (which has the lowest cost (resp. best QoS) does not meet the user QoS requirements.

5.1. Motivating example

Let us assume that the user asks to play a video news article with (color, 25 frames/S, TV Resolution) as QoS and 6 $ as a maximum cost (the desired and the worst acceptable values are the same). The QoS manager makes use of our negotiation procedure, and finds the following user offers (after mapping the corresponding system offers): (Color, 15 frames/s, TV Resolution) at 5$, (GREY, 25 frames/s, TV resolution) at 4 $, and (Color, 25 frames/s, TV resolution) at 6 $.

The QoS manager should be able to classify those three offers: which is the best offer and which is the worst offer? A simple solution is to present the possible offers to the user, and it is up to him/her to select one offer. However this solution has several drawbacks:

(1) Since we have to present the whole offers to the user, we cannot make any resources reservation. It does not make sense to reserve resources for all the offers. Then when the user selects an offer, we are not certain that the offer will be supported by the system, e.g. in the case of resources shortage; the user will not be happy with this situation, particularly if it happens frequently.

(2) Many offers may be produced for a given request, and it is not appropriate to present a large numbers of offers to the user. We think that the user, especially a novice, may become confused. Furthermore, the presentation of several offers will complicate the QoS GUI.

(3) The user may select an offer which does not make an optimal usage of system resources.

Given those drawbacks, we decided to define a classification procedure that allows to automatically classify the offers. Thus only one offer, for which the resources are reserved, will be presented to the user. The latter may accept or reject the offer, or initiate a renegotiation. Consequently when the user accepts an offer, he/she is certain that the offer will be supported by the system.

5.2. Classification procedure

We associate two parameters for each feasible offer: its static negotiation status and its importance.

5.2.1. Static negotiation status

The static negotiation status indicates the degree of satisfaction of the user profile by a given system offer. We have considered three values of the static negotiation status, however, more values may be considered:

(1) *DESIRABLE:* the QoS satisfies the QoS desired by the user;

(b) *ACCEPTABLE:* the QoS is better than the worst acceptable QoS values accepted by the user.

(c) *CONSTRAINT:* the QoS of the offer does not meet the worst acceptable QoS values requested by the user (for at least one monomedia and some of its characteristics).

Given a system offer, the computation of the static negotiation status consists of a simple comparison between the QoS associated with the offer and the user profile.

As an example, let us assume that (1) the user asks to play a news article with (color, TV resolution, 25 frames/s) as desired QoS and as the worst acceptable QoS, and 4 $ as the maximum cost to pay, and (2) the QoS manager produces, after the two first steps, the following offers (after the mapping activity):

- offer1: (black&white, TV resolution, 25 frames/s) with a cost of 2.5$,
- offer2: (color, TV resolution, 15 frames/s) with a cost of 4$,
- offer3: (grey, TV resolution, 25 frames/s) with a cost of 3 $, and
- offer4: (color, TV resolution, 25 frames/s) with a cost of 5$.

The QoS manager computes the SNS for each offer. The results are: offer1: *CONSTRAINT*, offer2: *CONSTRAINT*, offer3: *CONSTRAINT*, and offer4: *ACCEPTABLE*.

5.2.2. Importance

As mentioned above (see Section 3), the importance factors indicate the relative importance between QoS characteristics and cost. For some user, for instance, QoS is more important than cost, that is, he/she is willing to pay "any" cost to get the requested QoS; while for another user, cost is more important than QoS, that is, he/she is willing to accept a degraded QoS to not pay more than the maximum cost (specified in the user profile). The importance factors allow the user to specify his/her specific preferences (which are different from user to user) regarding QoS characteristics and cost.

(a) Computing the importance factors for QoS

For each QoS parameter, the user specifies the importance factors for only a specific set of values. For example, the user sets the importance values only for HDTV rate,

TV rate and frozen rate when dealing with the frame rate parameter. If the user selects a frame rate (specified in the user profile) different from these specific values, the corresponding importance factor is computed assuming that the importance increases (or decreases) linearly from frozen rate to TV rate, and from TV rate to HDTV rate. In the context of the news-on-demand prototype, the user specifies the importance factors, for a given QoS parameter, for the QoS parameter values described in Figure 2 (resolution: HDTV resolution, TV resolution, minimum resolution; audio quality: CD, telephone; color: super-color, color, gray, black&white; etc.)

To each QoS parameter value (specified in the user profile), corresponds an importance factor. The latter corresponds to the importance value specified by the user, if the QoS parameter value is one of those described in Figure 2; otherwise it is computed using a linear function (see above).

To compute the importance factor for a set of QoS parameters, e.g. instance of video QoS parameters, we have only to sum the importance values which correspond to the values of the QoS parameters.

We associate a default importance value for each QoS parameter value (in Figure 2). However, at any time during the negotiation phase, the user may modify these values to meet his/her goals. Indeed, for some user the video color parameter is more important than the video resolution, while it is the opposite for another user. The profile manager provides the user with facilities to set importance values for QoS parameters of interest. This provides the user with means to specify which are the media, e.g. audio or video, and QoS parameters of importance.

(b) Computing the importance factors for cost

Similarly, the user sets an importance factor for the cost. This importance factor indicates the importance of a cost of 1 $. A small importance factor indicates that the cost is not very important, while a high importance factor indicates that the cost is a very important parameter to consider. To compute the cost importance factor for an offer, we have only to compute the product of the importance factor that corresponds to 1 $ (set by the user; otherwise the default value is used) and the cost (to be charged to the user) which is associated with the offer.

(c) Computing the importance factors for a system offer

As mentioned above, the role of the QoS manager is to classify possible offers (configurations) to select an optimal one. Then we should be able to compute the overall importance factor, which will be used for this classification.

Given an offer, the following steps are performed to compute the overall importance (overall_importance) factor:

- compute the importance factor (QoS_importance) which corresponds to the QoS associated with the offer.
- compute the importance factor (cost_importance) which corresponds to the cost associated with the offer.
- substrate the cost importance from the QoS importance (overall_importance=QoS_importance - cost_importance).

The importance concept allows to compute a value for each offer including QoS and cost requirements, and thus allows to associate a value (of the same semantic) to any offer.

In order to satisfy the user requirements, it is not sufficient to classify the offers according to the overall importance factor (OIF). Indeed the QoS of the best offer (with the highest OIF) may be very different from the QoS requested by the user: The best offer is effectively the best in terms of cost/QoS, however it may not correspond to the user wishes. To overcome such a problem we use the static negotiation status as primary classification parameter, and the OIF as the secondary classification parameter. Thus the OIF allows to classify these offers which have the same static negotiation status.

We do not consider systematically the offers from the best offer to the worst offer: The best offer may have a cost higher than the cost the user is willing to pay. At first we consider only the offers which satisfy the cost and the QoS requested by the user. If none of those offers can be supported by the system (not enough available resources), we consider the other offers, however always in the order defined above.

Example

In this example we consider the offers described in the example of Section 5.2.1. To demonstrate the *impact* of the overall importance factor (OIF) on the classification of offers, we present the results of classification for three different settings of importance factors.

(1) Let us assume that we have the following importance factors: color: 9, grey: 6, black&white: 2, TV resolution: 9, 25 frames/s: 9, 15 frames/s: 5, and cost importance: 4.

The QoS manager computes the OIF for each offer and produces the following results: offer1: 10, offer2: 7, and offer3: 12, and offer4: 7.

Taking into account the static negotiation status (see the example of Section 5.2.1), the offers are classified as follows: offer4, offer3, offer1, and offer2.

(2) Let us assume that we have the following importance factors: color: 9, grey: 6, black&white: 2, TV resolution: 9, 25 frames/s: 9, 15 frames/s: 5, and cost importance: 0. This means that the cost is not an important constraint; the QoS is the main constraint.

The QoS manager computes the OIF for each offer and produces the following results: offer1: 20, offer2: 23, and offer3: 24, and offer4: 27.

Taking into account the static negotiation status, the offers are classified as follows: offer4, offer3, offer2, and offer1.

(3) Let us assume that we have the following importance

values: color: 0, grey: 0, black&white: 0, TV resolution: 0, 25 frames/s: 0, 15 frames/s: 0, and cost importance: 4. This means that the QoS is not an important constraint; the cost is the main constraint.

The QoS manager computes the OIF for each offer and produces the following results: offer1: -10, offer2: -16, and offer3: -12, and offer4: -20.

Taking into account the static negotiation status, the offers are classified as follows: offer1, offer3, offer2, and offer4.

6. QoS mapping

The user conveys his/her requirements in terms of parameters that are directly expressed in terms of human-perceptible quantities, e.g. CD quality for the audio (Section 3). Thus the parameters resulting from the user request should be transformed, by the QoS manager, to QoS parameters that the system can handle and manage. Examples of such parameters are delay, throughput, loss rate and jitter.

In the current prototype, from the QoS parameters values specified by the user, the QoS manager computes the parameters *maxBitRate, and avgBitRate* required to deliver the document.

The data is assumed to be stored as a suite of blocks, e.g. video frames and audio samples, on a disk. Concerning continuous data, the size of blocks assumes a value between a maximum and a minimum length. This length depends on the compression scheme used to compress the original data and the nature of the data itself. The block length, namely the maximum and the average length, of a monomedia of the document, is stored in the MM database [Vit 95].

If the data is sent to the user without transformation, as in the case for our prototype, the throughput is computed as follows.

For video:

maxBitRate = (maximum frame length) *(frame rate),

avgBitRate = (average frame length) *(frame rate).

For audio

maxBitRate = (maximum sample rate)*(sample rate)

avgBitRate = (average sample rate)*(sample rate)

Concerning the other parameters, such as jitter (which is compensated by synchronization protocols [Lam 94]) and loss rate, we use specific values for video and audio presented in [Ste 90] based on some experiments. As an example the following values are considered for the video: jitter= 10 ms, and loss rate 0.003.

7. Cost computation

The cost concept is key in any QoS negotiation procedure. The cost will limit the greediness of the users. Without cost constraints, the users will ask for the best QoS available, increasing the blocking probability of the system since the amount of system resources is finite. Consequently only a few users will be accepted to use the service and thus the system will be obliged to increase the cost of its services to get at least the money required to use the resources. Consequently only the wealthy users can make use of the services provided.

The cost depends on several factors, such as the requested QoS, the duration of the session, the type of guarantees, and the type of service. In the current prototype the cost is computed as the sum of the server cost, the network cost, and cost related to the document, e.g. copyright cost. The main QoS parameter, which we consider in the cost computation, is the throughput, and the type of guarantees, e.g. best-effort or guaranteed service.

To compute the network cost, we assume the existence of a cost table which stores the cost (per time unit) for each value of throughput. Since it is not possible to consider all possible values of throughput (infinite list), only a range of throughput classes are considered. Similar tables are used to compute the cost to use the server resources for accessing the MM document.

For each monomedia component of the document to be played, we compute the network cost and the server cost using the corresponding cost tables. If the length of the monomedia M_i to be played is D_i, the throughput belongs to the throughput class $C_{i^\dagger}$, and the network (resp. server) cost which corresponds to $C_{i^\dagger}$ is $CostNet_{i^\dagger}$ (resp. $CostSer_{i^\dagger}$), then the network cost to deliver M_i is $CostNet_i = CostNet_{i^\dagger} \times D_i$ (resp. $CostSer_i = CostSer_{i^\dagger} \times D_i$). The cost, $CostDoc$, to be charged to the user is computed as follows:

$$CostDoc = CostCop + \sum_{i=1}^{n} (CostNet_i + CostSer_i) \quad (1),$$

where $CostCop$ indicates the document copyright, and n the number of monomodia of the document.

8. QoS user graphical interface (QoS GUI)

The QoS GUI (see Section 3), allows the user to negotiate and renegotiate his/her requirements in terms of QoS/cost. The profile manager is responsible for supporting these facilities.

Most of the windows that are displayed on the screen by the profile manager have been generated using AIC version 1.2. AIC is a software that is used to build GUI's. It allows to create, generate and test code for any graphical user interface in a user friendly manner. However, AIC does not support dynamic generation of the contents of menus. The

AIC-built windows are the one that are generated by the profile manager by opposition to the windows generated by the media players, which have been implemented directly using X Window or Motif programming.

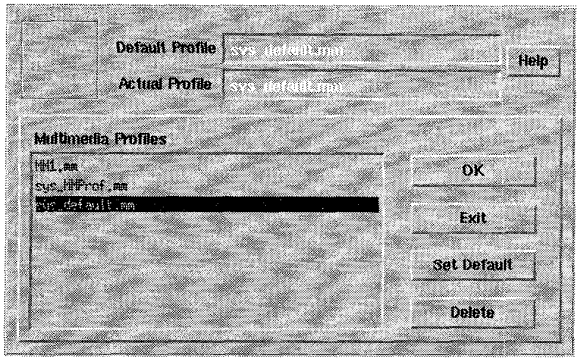

Figure 3. Main window

The main window (Figure 4) of the QoS GUI appears as soon as the user pushes on *Play with QoS* or *Open Window* in the news-on-demand prototype user interface [Vel 95]. The main function of this window is to allow the user to select, edit or delete a user profile, or to set a default user profile. If the user wants to leave the QoS GUI, he/she must push *EXIT*. When the user selects the desired user profile, he/she clicks on *OK* to start negotiation. The callback function causes the QoS manager to start executing the QoS negotiation procedure.

If the user double-clicks on a given user profile, the profile component window (Figure 5) appears displaying the list of monomedia, time, and cost profile (see Section 3). The profile components of the user profile selected are highlighted. To modify or create a new user profile, the user selects the profiles of interest (monomedia, time, and cost profiles) and pushes on *Save* or *Save as* respectively.
The profile component window appears also when the negotiation fails. In this case the constraint buttons of the profiles, which cannot be satisfied by the system, are activated with red color.

If the user wants to edit a profile, he/she must double-click on this profile and the corresponding profile window appears (Figure 6). Each profile, namely video, audio, text, image, time, and cost profiles, has an associated customized profile window which allows the user to specify his/her requirements. Through scaling bars, and predefined values the user is able to set the values for QoS parameters, cost, time and importance. For each QoS parameter the user is able to set the desired value and the minimum value acceptable.

If the constraint button corresponds to the profile is red (see profile component window), the offer provided by the system is also displayed for each QoS parameter (on the offer scaling bar). If the user accepts the offer, he/she should push *OK*; otherwise he/she should push *CANCEL* to reject the offer, or modify the offer and then push *OK* to initiate a renegotiation.
The profile modification and creation facilities are provided by the *Save* and *Save as* buttons, respectively. If the user wants to play a stored locally example with the current profile he/she may push on *show example*. The callback function causes the MPEG player to play a monomedia example which satisfies the current profile.
At any time the user is able to cancel his/her profile settings using the button *CANCEL*.

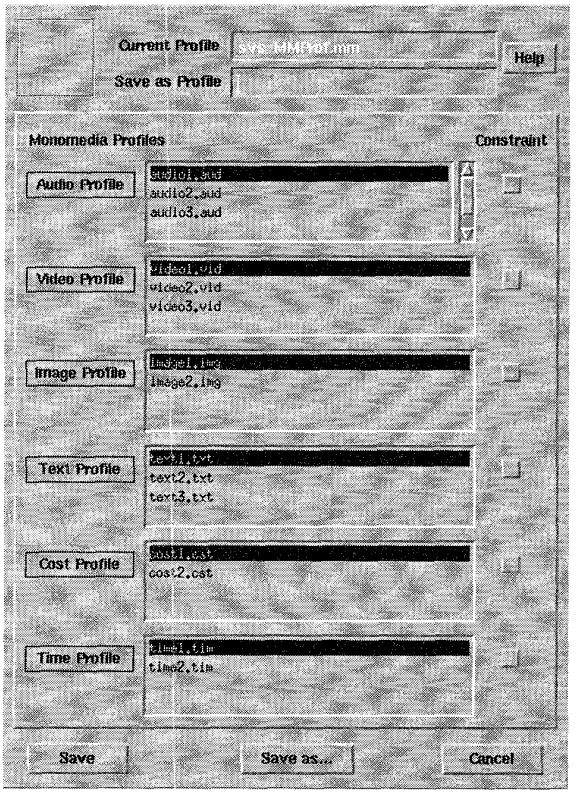

Figure 4. Profile component window

The profile manager uses the information window (Figure 7) to display the results of the negotiation process. If the negotiation failed, the profile manager displays the negotiation status, namely FAILEDTRYLATER or FAILEDWITHOUTOFFER; otherwise the profile manager displays the values of the different QoS parameters and the corresponding cost. The user should press OK, within a given amount of time, to start the delivery of the article. A timer is initialized to a value *choicePeriod* and started at the time of the window is displayed. If a time-out is reached before pressing *OK*, the session is simply aborted and a new negotiation is required if the user wants to play

the article.

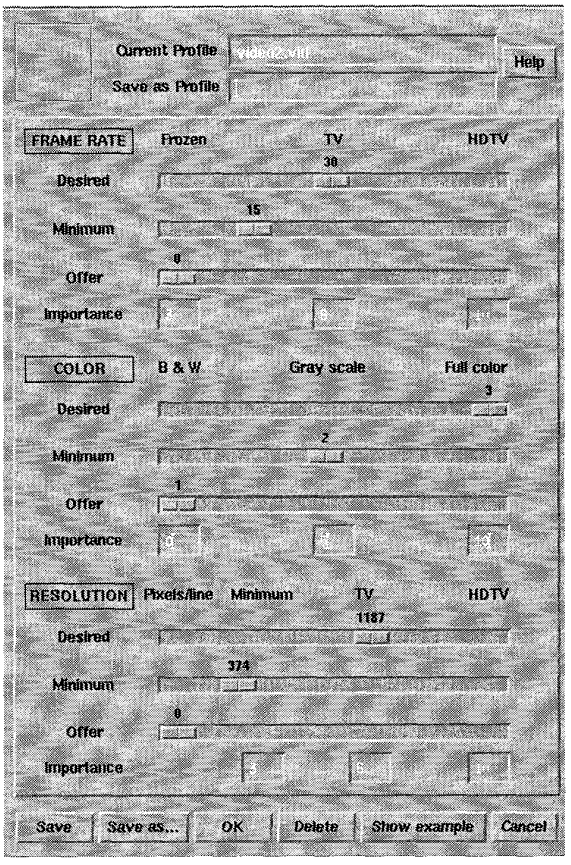

Figure 5. Video profile

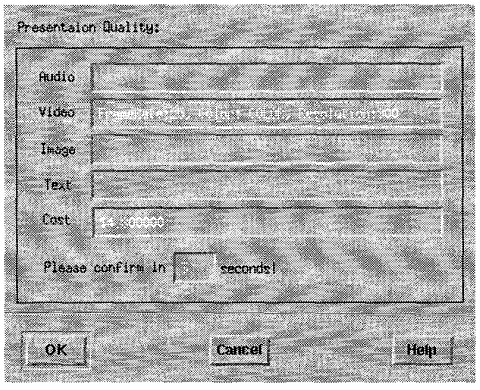

Figure 6. Information window

8. Conclusion

In this paper we concentrated on QoS management aspects, namely QoS (re-) negotiation and adaptation, for presentational MM applications. A new QoS negotiation procedure, based on a general framework for negotiation [Haf 95c], has been designed and implemented. The procedure makes use of an *optimization approach* to find an optimal component configuration which supports the user preferences in terms of QoS, and cost. We defined (1) two parameters, namely static negotiation status and the overall importance factor, which are used to classify the possible configurations from the "best" one to the "worst" one, and (2) techniques to compute the values of these parameters for MM documents. The proposed procedure enables *smart negotiation*, in opposition to *basic negotiation* provided by the existing QoS architectures, which increases the availability of the system and the user satisfaction. Furthermore, it supports the negotiation of a multimedia object (audio and video) as an atomic object, that is, the optimization is performed taking into account all monomedia components of the document at the same time, while existing approaches concentrate on the negotiation of a single monomedia object. More generally the procedure can be used for negotiation, renegotiation, and adaptation with almost no modifications.

The QoS GUI is customized in our implementation for a range of presentational applications; however it can be used for any application handling MM information, such as teleconferencing systems. There are several window layers which allow the management of user profiles, such as creation or modification. The user profiles may include further QoS and cost preferences of the user, other information related to document search, e.g. the user prefers certain servers over others, security, etc.

The implementation has shown the practicality the our proposals related to QoS management, and has demonstrated that the provision of *smart negotiation* and *automatic adaptation* are viable features for presentational MM systems and improve their usability.

Acknowledgment

We would like to thank our colleagues of the CITR QoS subproject for their contribution to the work presented here. Special thanks go to Quoc Vu, Benjamin Isnard and Jun Zhang who participated in the implementation of the QoS GUI. This work was supported by a grant from the Canadian Institute for Telecommunication Research (CITR), under the Networks of Centres of Excellence Program of the Canadian Government.

References

[Boc 96] G.v.Bochmann, B.Kerherve, A.Hafid, P.Dini and A.Pons, *Functional Design of Adaptive Systems*, To Appear in the IEEE Workshop on Multimedia Software Development, Berlin, Germany 1996

[Cam 94] A.Campbell, G.Coulson and D.Hutchison, *A Quality of Service Architecture*, ACM Computer Communication Review,

April 1994

[Dub 95] E.Dubois, N.Baaziz and M.Matta, *Impact of scan conversion methods on the performance of scalable video coding*, SPIE 95

[Fer 95] D.Ferrari, *The Tenet Experience and the Design of Protocols for Integrated Services Internetworks*, Multimedia Systems Journal, November 1995

[Gop 94] G.Gopalakrishna and G.Parulkar, *Efficient Quality of Service in Multimedia Computer Operating Systems*, Department of Computer Science, Washington University, Report WUCS-TM-94-04, 1994

[Haf 95a] A.Hafid and R.Dssouli, *A Negotiation Model for Distributed Multimedia Applications*, in Proceedings of the first International Conference on Multimedia Networking (IEEE Computer Society), Aizu-Wakamatsu, Fukushima, Japan, September 1995

[Haf 95b] A.Hafid, *A Hierarchical Negotiation for Distributed Multimedia Applications in a Multi-Domain Environment*, In Proceeding of the Second International Workshop on Protocols for Multimedia Systems, Salzburg, Austria, 1995

[Haf 95c] A.Hafid, G.v.Bochmann, B.Kerherve, R.Dssouli and J.Gecsei, *On QoS Negotiation in Distributed Multimedia Applications*, Submitted to IEEE JSAC

[Haf 95d] A.Hafid and G.v.Bochmann, *Quality of Service Adaptation in Distributed Multimedia Applications*, submitted to Multimedia Systems Journal (A Short version paper has been accepted in IEEE/IFIP ICDP96, Berlin, Germany, 1995)

[Haf 96] A.Hafid, G.v.Bochmann and R.Dssouli, *Quality of Service Negotiation with Future Reservations: A detailed Study*, Submitted to Computer Networks and ISDN Systems Journal (A short version is in Proceedings of the Fourth IFIP International Workshop on Quality of Service, Paris, France, 1996)

[Ker 95] B.Kerherve, A.Pons, G.v.Bochmann and A.Hafid, *Metadata Modelling for Quality of Service Management in Distributed Multimedia Systems*, To Appear in Proceedings of IEEE Conference on Metadata, 1995

[Kes 95] S.Keshav and H.Saran, *Semantics and Implementation of a Native-Mode ATM protocol Stack,* Internal Technical Memo, AT&T Bell Lab, Murray Hill, NJ, January, 1995

[Lam 94] L.Lamont, *Component Interactions and Messaging System for Multimedia Synchronization*, Technical Report, MCR-Lab University of Ottawa, May 1994

[Nah 95] K.Nahrstedt, *An Architecture for End-to-End Quality of Service Provision and its Experimental Validation,* Ph.D. Thesis, University of Pennsylvania, 1995

[Neu 96] G.Neufeld, D.Makaroff and N.Hutchison, Design of a Variable Bit Rate Continuous Media File Server for an ATM Network, SPIE96, 1996

[Vit 95] C.Vittal, M.Ozsu, D.Szafron, G.Medani, *The Logical Design of a Multimedia Database for a News-On-Demand Application*, Technical Report #94-16, The University of Alberta, Canada, 1994

[Vel 95] R.Velthuys, J.Wong, K.Lyons, G.v.Bochmann, E.Dubois, N.Georganas, G.Neufeld, T.Ozsu, *Enabling Technologies for Distributed Multimedia Applications*, CITR Internal Report, 1995

[Vol 95] C.Volg, L.Wolf, R.Herrtwich and H.Wittig, *HeiRat - Quality of Service Management for Distributed Multimedia Systems*, Multimedia Systems Journal, November 1995

[Zha 95] L.Zhang, et al., *RSVP Functional Specification,* Working Draft, draft-ietf-rsvp-spec-07.ps, 1995