On Distributed Multimedia Presentational Applications:
Functional and Computational Architecture and QoS Negotiation*

B. Kerherve[b], A. Vogel[a],
G. v. Bochmann[a], R. Dssouli[a], J. Gecsei[a] and A. Hafid[a]

[a]Université de Montréal, Dpt. d'IRO,
CP 6128, Succursale Centre Ville, Montréal, H3C 3J7, Canada

[b]Université du Quebec a Montréal, Dpt. de Mathematiques et d'Informatique,
CP 8888, Succursale Centre Ville, Montreal, H3C 3P8, Canada

Abstract

Approaches to a functional and a computational architecture for distributed multimedia presentational applications are developed. These approaches are illustrated by a case study, a multimedia news-on-demand service. The concept of Quality of Service (QoS) parameters is seen to determine the new characteristics of distributed multimedia applications. Accordingly the computational architecture for distributed multimedia presentational applications is developed in a QoS driven way within the the framework of the Reference Model of Open Distributed Processing. The concept of QoS interfaces is introduced in order to handle the QoS negotiation in a general and generic way. Objects in a distributed multimedia application can negotiate their QoS parameters through these QoS interfaces. Using this approach, variants of QoS negotiation protocols are investigated.

Keyword Codes: C.2.2; H.5.1
Keywords: Network Protocols, Multimedia Information Systems

1. Introduction

The new quality of distributed multimedia applications is characterized by handling continuous media, e.g. audio and video, and by managing various medias at the same time. Distributed multimedia applications can be classified by presentational, conversational, or having both aspects. Presentational applications provide remote access to multimedia documents. Examples are news-on-demand [18] or video-on-demand [28] services. Conversational applications involve multi-directional, real-time, multimedia communications, e.g. video conference [26] or collaboration services [1]. Systems for distance education have both aspects.

In the framework of our project [34], we focus on distributed multimedia presentational applications, a multimedia news-on-demand service is selected as a case study. The

* This work was supported by a grant from the Canadian Institute for Telecommunication Research (CITR), under the Networks of Centers of Excellence Program of the Canadian Government.

purpose of the news-on-demand system is to offer an integrated, computerized multimedia news service to various customers. The contents is extracted from existing news sources such as radio, TV, wire services, print, and re-composed as (possibly personalized) multimedia objects which the clients will access. The documents contain images, text, audio and video. Potential users are government institutions, decision makers in companies, journalists, business services, etc. The system runs on a fully distributed architecture where multimedia data are stored on different sites, and users can access them from different places through the network. Functional architecture for a news-on-demand system is investigated in Section 2.

In general, multimedia applications within an open environment, i.e. heterogeneous end-systems and various types of networks, require heavily on considerations of quality of service (QoS) parameters. The connection of two end-systems by a network in the sense of the 'traditional' OSI model does not guarantee the inter-operability of the end-systems. Additionally, matching interfaces are required and within these, QoS parameters are an important factor, e.g in the case of a video-on-demand service, all the three partners, producer, network, and consumer, have to be able to process e.g. the same data rate; if the producer and the network are too fast for the consumer, the service becomes senseless. We will address this question within the Reference Model on Open Distributed Processing (RM ODP) [27]. Using the ODP concepts, a computational architecture for distributed multimedia presentational applications is developed and applied to the news-on-demand service in Section 3.

In order to clarify the responsibilities and interrelationships of the various objects and their QoS parameters, we propose in Section 4 a QoS architecture by introducing the concept of QoS interfaces. Within this architecture, mechanisms for the QoS management have been developed for our focused case study which are introduced in Section 5. Generalizations of our approach are discussed in Section 6 which also concludes the article.

2. Functional architecture

The news-on-demand system offers different functionalities to the users. In this section we first identify the different classes of users, we then present the functional architecture of the news-on-demand system and lastly we describe the different user interfaces provided by the system.

The users involved in the process of production and consumption of multimedia news can be classified into four categories: news *publishers*, news *analysts*, news *producers* and news *consumers*.

News publishers are responsible for delivering the news materials over different media and for associating descriptive and indexing information to these objects. The role of news analysts is to associate semantic information to the different monomedia objects such as key words or specific interpretation of the object. News producers build multimedia objects by linking monomedia elements and by producing control information such as presentation formats (templates) and synchronization scenarios to define the temporal order of display. The multimedia news consumer inspects the database in order to find multimedia news of interest to him. He can consult the multimedia news database in different ways: browsing, specifying an identifier of a news object or by conditional queries. These consultation modes offer a wide range of possibilities to explore the database.

The Figure 1 proposes a general functional architecture for multimedia presentational applications. This architecture is composed of three autonomous levels: the database level, which is the system's nucleus, the function level and the user interface level which can be considered as shells defined on top of the system's nucleus. The user interface shell is in charge of managing the different user interfaces which answer the specific user's needs. This level is also concerned with the user's environment, it offers services to set, modify or delete a specific user profile and takes into account the parameters of the quality of service. The function shell is responsible for the transformation of a user's specific demand to the target operations required on the multimedia database. The database level is responsible for the storage and the access to the multimedia database.

In the news-on-demand system, different user interfaces can be identified as shown within the Medialog project [6]. We define the following three different user interfaces: *monomedia news storage and analysis*, *multimedia news consultation* and *multimedia news production*. They answer specific needs of various user groups and consider the environment associated to these groups. The news-on-demand service users define their favorite environment such as the word processor, the communication environment, the parameters of quality of service as well as the tools for document annotations. This environment can be dynamically modified.

Figure 1 also shows the functional architecture for the news-on-demand system and distinguishes the different user interfaces and functions that are presented in the next three sections.

2.1. Monomedia News Storage and Analysis

The monomedia storage and analysis user interface is dedicated to the news publishers who load the database with basic monomedia objects coming from a single medium, e.g. radio, television, newspaper. This user interface offers connections to specific equipment (video camera, sound recorder, digitizers...) to capture these objects. After being captured and digitized, the objects are described and analyzed by monomedia analysts. The analysts associate registration information to the objects as well as description information describing their contents. Each medium has a predefined schema for the description of registration and description information. Generally these schemata contain keywords and a brief description.

The monomedia news storage and analysis user interface interacts with storage and analysis function, which performs three different tasks: monomedia objects *storage*, monomedia objects *description*, and monomedia objects *indexing*.

The storage and analysis function identifies the internal structures that are required for the database storage of the monomedia objects and executes the transformation of the external structure into these internal structures.

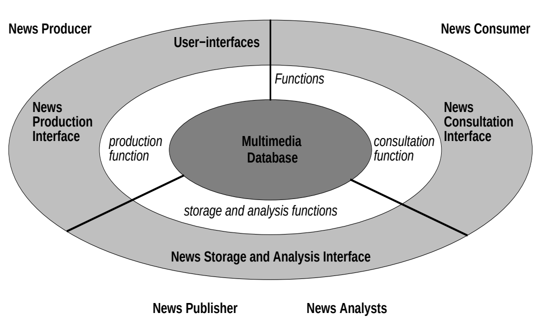

Figure 1. Functional architecture for the news-on-demand system.

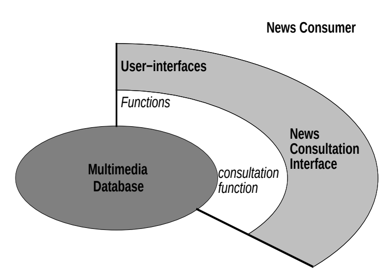

Figure 2. Considered functions of the news-on-demand system.

For example, if the HyTime standard [22] is the external structure and if an object oriented database is used, the storage and analysis function will identify the different classes which are concerned with this document and will generate the corresponding requests to be submitted to the database system. This transformation step will be different if the database system is an extended relational database system.

The storage and analysis function therefore generates the database manipulation language requests for the insertion of monomedia objects and is also responsible for the efficient storage of the monomedia objects, that means that this component of the system decides whether specific compression or indexing techniques can be used.

In order to be efficiently retrieved, the monomedia objects have to be described and classified. This description is done according to specific description schemata that are stored in the database. The storage and analysis function uses these schemata to provide the user interface level with the description information that can be used in the description and classification process.

2.2. Multimedia News Consultation

The final users of the news-on-demand service search multimedia documents in the system and can reproduce these documents. The system provides reproduction facilities and manages the copyright fees. The consultation of multimedia news documents can be separated into two different and complementary steps: the *search process* and the *display and manipulation process*.

The display and manipulation process follows the search process and offers the user the possibility to display the document and to process specific actions on the retrieved object.

In order to search multimedia documents, the users can use several retrieval techniques [2]: retrieval by keywords, browsing, guided tour, full text retrieval, similarity retrieval. Several techniques such as retrieval by keywords, browsing or guided tours are general for all the media while others, such as similarity retrieval or full text retrieval, are specific for still images or texts. In [15] we present in detail the consultation user interface which offers these different possibilities and their combination to the user. At the end of the searching process, the user can execute some operations on the document. These operations are independent and not linked to the search process. The consultation user interface must offer tools to facilitate this kind of operations. Four different tasks can be identified for the consultation function: access to the database system for retrieval functions, retrieval workspace management, internal to external model transformation, and temporal and spatial relationships enforcement.

All the information concerning the database access is transmitted to the consultation function, which generates the corresponding request formulated in the database manipulation language. At this step, the consultation function may enrich the user's requests based on his profile. The requests are then submitted to the database system and the consultation function waits for the results to send them to the consultation user interface. At this step the consultation function may keep some parts of the result in the workspace for further use. The efficient management of this workspace is an important task to be performed by the retrieval function.

While transmitting the results to the user interface level, some transformation may be required to change from the internal model supported by the database system to the external model required by the user interface level. This task is symmetrical about the one performed by the storage and analysis function. The last task performed by the retrieval function is the enforcement of spatial integration and temporal synchronization of the different components of the multimedia news document. Actually, spatial and temporal relationships between the components are an important characteristic of the multimedia news document defined through the production user interface. These relationships are stored in the database as part of the presentation component of the multimedia news document. The retrieval function is in charge of extracting these relationships from the document and to enforce them while interacting with the user interface level.

2.3. Multimedia News Production

The multimedia news production user interface is dedicated to the generation of multimedia news documents. This user interface is concerned with the creation of documents as well as with their modification. The creation of a multimedia news document needs the specification of the following three components [7,17] of the document: the *structure*, the *content* and the *presentation*.

The structure defines the objects that are part of the document and their organization, the content contains these objects (a set of monomedia objects of different types) and the presentation gives information about the layout and the synchronization of the document. The presentation part of the document specifies how the components of the multimedia news document will be presented to the final user. This specification is made through temporal and spatial relationships between objects. The temporal relationships describe the synchronization points while spatial relationships describe the organization of the components on the screen. These relationships define constraints that will be enforced by the consultation function driving the display of the document.

While defining the content of the multimedia document, the user might choose to reuse existing documents or to capture new objects. The reuse of existing documents needs a preliminary step to retrieve them, while capturing new objects such as audio or video needs the connection to specific equipment. The production user interface will therefore use the same services as the consultation user interface and the monomedia storage and analysis user interface.

Since the construction of multimedia news documents is a difficult task which can be progressively refined, the system must offer facilities permitting to modify a document. The modifications can be made on the structure, the content or the presentation part of the document.

When the user defines the components, the multimedia news production user interface captures all the information, whether with specific equipment for images, sound or video, or with specific tools such as word processors. All the elements are defined corresponding to an external multimedia document model, captured, and then transmitted to the production function for database storage. Like the storage and analysis function, the production function is also responsible for transforming the external structure to the internal structure supported by the database system and thus will share services with the former function.

In the second case, when a user builds a multimedia news document from an existing one, the production user interface connects him to a subset of the consultation user interface in order to select the pertinent documents. The production function must keep the references to those documents in a workspace in order to use them in the production process. Since the system must allow object sharing, those references will be incorporated in the new document using a composition mechanism. The production function is in charge of this document composition mechanism. It is also responsible for transforming the structure of already existing documents into other structures if they are offered at the external level.

3. Computational architecture

In this section a computational architecture for distributed multimedia presentational applications is developed and applied to the news-on-demand service. We will present an architecture within the framework of the Reference Model of Open Distributed Processing (ODP). ODP and the corresponding Reference Model was characterized by Kerry Raymond as follows [27]: ''Advances in computer networking have allowed computer systems across the world to be interconnected. Despite this, heterogeneity in interaction models prevents interworking between systems. Open Distributed Processing describes systems that support heterogeneous distributed processing both within and between organizations through the use of a common interaction model. ISO and ITU-T (formerly CCITT) are developing a Basic Reference Model of Open Distributed Processing (RM ODP) to provide a coordinating framework for the standardization of ODP by creating an architecture which supports distribution, internetworking, interoperability and portability.''

The main concept of the RM ODP is the viewpoint. A viewpoint is an abstraction, focused on parts of an ODP system determined by a particular interest. There are five viewpoints: *enterprise, information, computational, engineering,* and *technology*. They are defined by corresponding description languages. An introduction using a multimedia example is given in [33]. We consider here the computational viewpoint which is a functional decomposition of the system into objects that are candidates for distribution. Hence, the corresponding computational language specifies the system in terms of communicating objects. Computational objects are providing operations through computational interfaces. Interaction between computational objects is described in terms of interface binding.

For distributed multimedia presentational applications we identified two main objects: a *server* containing multimedia documents, and a *client* providing access to the server's documents. The server object is composed of several sub-objects, e.g. for different types of storage objects. When accessing a document, the client and the server will perform the roles of a producer and consumer, respectively. Both objects have several interfaces: *operational interfaces* providing operations for retrieval, access and control and *stream interfaces* supporting the continouos data transfer. From the computational viewpoint, communication aspects are only visible in terms of interface binding and quality of service parameters.

In the following we will develop a computational specification of a news-on-demand service. However, only the consultation part of the complete service, as shown in Figure 2, is specified. Figure 3 illustrates the computational viewpoint of the remaining news-on-demand service. It contains two main computational objects, a multimedia server (MM-Server) and a client (MM-Client). The server is thought to manage the different types of multimedia storages. The multimedia server is composed of sub-objects. We identified the following ones: the *database (DB) server, continuous media (CM) file servers, noncontinuous media (NCM) file servers*, and *archival storages*.

A client has initially three operational interfaces, called the search, the access and the QoS negotiation interface. The search interface provides operations (according to the search process described in Section 2.2) to retrieve information from the database, e.g. retrieve by keywords, browsing, guided tour, full text retrieval, and similarity retrieval. The access interface provides operations (according to the display and manipulation process described in Section 2.2) to access and control the access to a multimedia document, e.g. start, stop, fast-forward. The QoS negotiation interface provides operations supporting the QoS negotiation and renegotiation. On demand, stream interfaces will be created to access to a multimedia document. Such a scenario is shown in Figure 3.

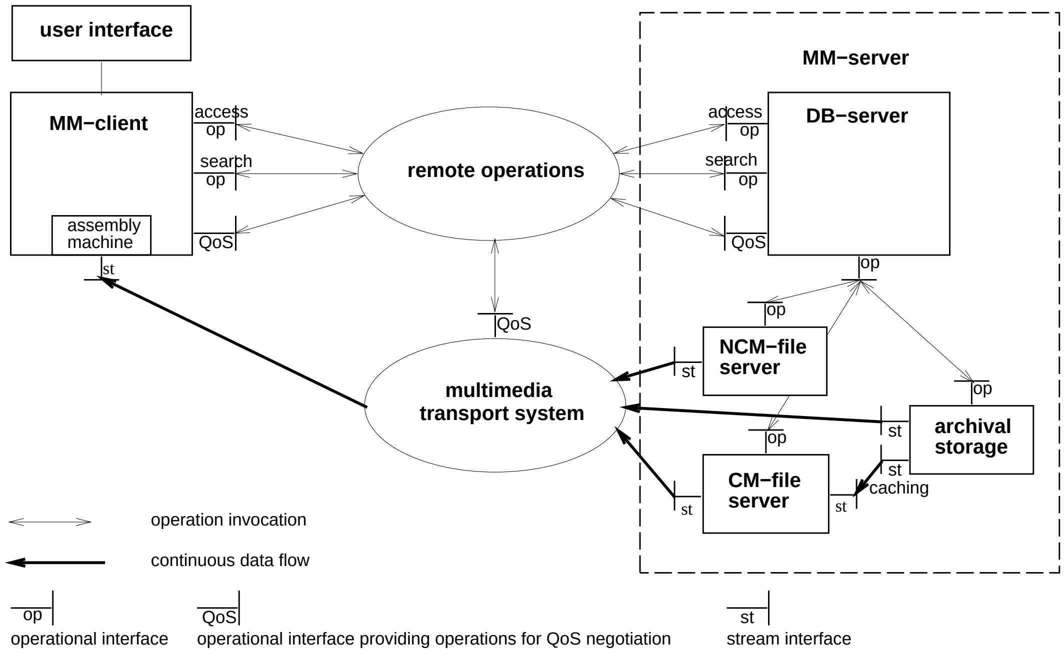

Figure 3. Computational specification of the multimedia news-on-demand service.

An incoming data stream has to be processed in order to be displayed in the appropriate way. This internal processing is indicated by the box called *assembly machine*. It includes the processing for synchronization of different streams, e.g. by the media synchronization controller [16], or processing of specialized transfer formats, e.g. MHEG-documents [25].

The server object has the operational interfaces matching the client's ones. Multimedia objects require special purpose storages because of their properties e.g. immense size (even when compression techniques are employed) and continuous character, and the enormous number of objects. Consequently, the database server is holding instead of the multimedia objects itself only references to them. According to certain administration policies, news and archived information are distinguished and intended to be stored on different objects. The objects (shown in Figure 3) called continuous media (CM) file server, noncontinuous media (NCM) file server, and archival storage are supposed to contain recently published documents in continuous and non-continuous formats and archived documents, respectively. From a technology point of view, they are implemented by conventional databases, special purpose file servers and tertiary storages, e.g. juke boxes containing CDs or tapes, respectively. From the computational viewpoint, the technology decisions are described in terms of QoS parameters.

The quality of service parameters are investigated in Section 4 in more detail. Communication aspects are reduced to the binding of interfaces within the computational viewpoint. However to illustrate the binding, we added place-holders in Figure 3. The term *remote operations* stands for the binding of operational interfaces and *multimedia transport system* for the binding of stream interfaces.

4. QoS architecture

The term QoS architecture has been limited in most previous work to considering the QoS mapping between different layers in a protocol stack. The QoS parameters concerning protocols have been investigated in the Internet community [11] as well as in the OSI community [8]. This includes research on the mappability of QoS parameters of different layers [9,5] and the mechanisms to satisfy QoS requirements, i.e. mostly resource reservation [32,23]. Also operating systems support [12,3] for multimedia applications is under research.

However, less work is done on the application level where all the various QoS parameters play together. In this article we address this problem and try to approach it within the framework of Open Distributed Processing. Similar considerations, but closer to the transmissions aspects, were done at Lancaster [5]. Our view includes also the characteristics of operating systems and particular application objects, such as databases.

For distributed multimedia presentational applications we identified three main objects which are involved in the QoS management: the MM-server, the MM-client, and the multimedia transport system, as shown in Figure 4.

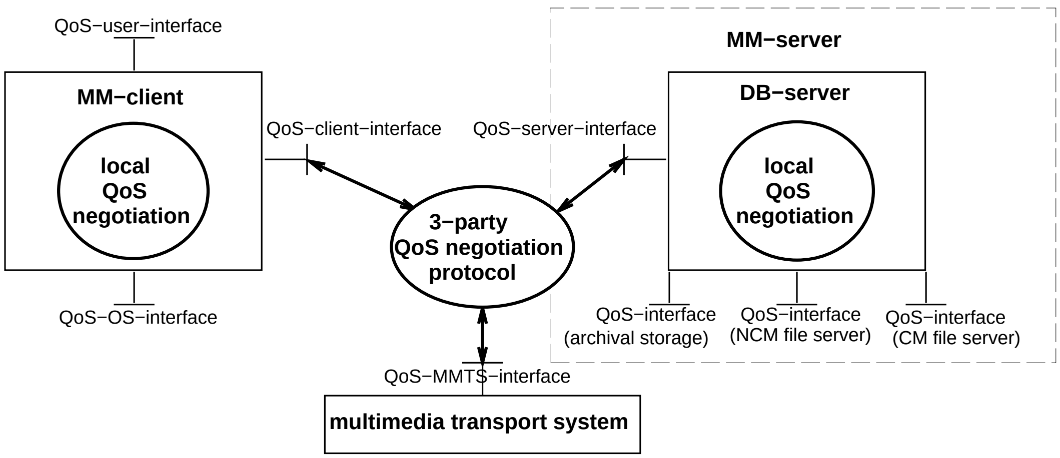

Figure 4. QoS management.

QoS interfaces are defined by tuples. The structure of the tuples follows a unique schema, a list of pairs of QoS parameters and their values. The possible values for a particular parameter are expected to be partially ordered. This assumption allows to use expressions for the value, e.g. ‘frame rate greater or equal then 25 frames/sec’. Also there are two predefined values for all parameters, namely ‘undefined’ and ‘any-value’.

An important concept which is used in the client and the server is the definition of the type of a multimedia object which we call media-type. We identified atomic and composed types. Atomic types are either mono-media types, e.g. text, images, audio or video; or multimedia objects which are multiplexed, e.g. audio and video in formats like DVI or MPEG. Composed types are constructed using atomic types as building blocks, however, additional information is required to describe how to compose them according to temporal and spatial relations. [16,31].

The following subsections consider QoS interfaces for the following objects:

- client: end-user (represented by the user interface) and operating system,
- server: database server, file server (continuous media and noncontinuous media), and archival storage
- transport system

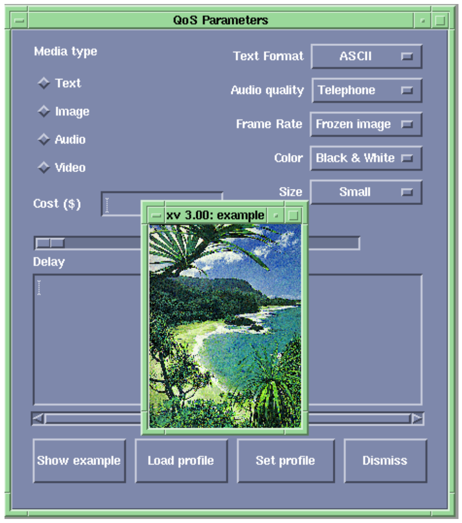

Figure 5. Quality query by example.

4.1. Client

Users-QoS interface

We suppose that a user is not willing to deal with raw numbers. We agree principally with ‘quality query by example’ [14]. However, our approach is to present the user a set of predefined parameters which corresponds to his/her experiences, e.g. audio quality known from the telephone or the compact disc. The user can check the offered qualities by listening to or watching examples which are provided for all QoS variants. A prototype of such an user interface, also called QoS demonstrator, is shown in Figure 5.

Table 1 shows the list of identified parameters. (We note that the possible values, mentioned in this in the foloowing tables, are only examples.)

Table 1
QoS parameters at user-interface

parameter type	possible values
media-type	{ text, image, audio, video, audio&video, composed-types }
text-format	{ ascii, postscript }
audio-quality	{ telephone, cd }
color	{ black&white, gray, color, super-color }
size	{ small, medium, large }
frame-rate	{ TV-rate,reduced-rate, frozen-images }
delay	integer (seconds)
cost	integer ($)

The actually defined tuple types are for the various medias:

<text, text-format, color, delay, cost> <image, color, size, delay, cost>
<video, color, frame-rate, size, delay, cost> < composed-type, skew, t_1, ..., t_n >

where skew is a skew or synchronization parameter and t_1, ..., t_n are QoS tuples of atomic media-types. Using these convention, an example is:
< media-type: audio&video, audio-quality: telephone, color: gray,
frame-rate: TV-rate, size: medium, delay: 1sec, cost: any >

Operating-system-QoS interface

The operating system QoS interface defines the quality of the support from the operating system. We identified two classes of QoS parameters abstracting from lower level properties such as CPU-power, memory-management and process scheduling. *Quality of devices*: including output devices, e.g. screen, loudspeaker, and storage devices used for buffering, e.g. hard disk. *Quality of software* to support the output of various medias, e.g. postscript-viewer or MPEG-player. The QoS parameters media-type, format, throughput, and guarantee class are identified. This leads to the parameters proposed in Table 2.

Table 2
QoS parameters at operating system interface

parameter type	possible values
screen	{ 1-bit, 8-bit-gray, 8-bit-color, 24-bit-color}
audio-device	{ telephone-quality, cd-quality }
disk storage	integer (bytes)
media-type	{ text, image, audio, video, audio&video, composed-types }
supported format	{ ASCII, postscript, gif, tiff, MPEG }
throughput	integer (frames/second)
delay	integer (seconds)
guarantee class	{ guaranteed, best-effort }

We take as an example the MPEG-player [29] running on a machine with a non-realtime operating system and 20 Mbyte available disk space, 8-bit-gray screen and a telephone-quality audio-device. A throughput of 10 frames/sec and a delay: 1 sec have been measured, but can not be guaranteed.
< screen: 8-bit-gray, audio-device: telephone-quality, disk storage: 20 Mbyte,
media-type: video, supported format: MPEG, throughput: 10 frames/sec,
delay: 1 sec, guarantee class: best-effort >

4.2. Server

Server-QoS interface

The server provides the access to and provides information on multimedia documents. The multimedia documents are characterized by their media-type, format, and size. Additional information is available from the storage objects which actually contain the multimedia documents. There is a throughput parameter which can be provided by a storage device to access an object. With respect to the throughput, the size of data packages, e.g. audio or video frames, is of interest. When accessing a stored object, there are also a certain delay and jitter. The delay can range from very small values, e.g. when reading from a hard disk, to quite large value, e.g. when data has to be loaded from a tertiary storage into a cache or data is preprocessed, for example converted from one format into another. A guarantee class parameter characterizes the throughput, the delay, and the jitter values with values best-effort or guaranteed. From the financial point of view, there are costs, in particular for accessing the database and for copyrights. This leads to the QoS interface as given in Table 3. According to the hierarchies shown in Figure 3, QoS parameters can be obtained dynamically from the lower layer objects using similar QoS interfaces from the tertiary storage and continuous media file server.

Table 3
QoS parameters at server-interface

parameter type	possible values
media-type	{ text, image, audio, video, audio&video, composed-types }
format	{ ascii, postscript, gif, JPEG, MPEG, DVI, ... }
color	{black&white, gray, color, super-color}
audio-quality	{ telephone, cd }
frame-rate	integer (frames/second)
size	bytes
throughput	integer (bytes/second)
delay	real (seconds)
packet size	integer (bytes) or variable
jitter	real (seconds)
guarantee class	{ guaranteed, best-effort }
cost	integer ($)

Example:
< media-type: video, format: MPEG, size: 12,345,678 byte, throughput: 0.5 Mbit/sec, delay: 15 sec, packet size: 50 kbyte, jitter: 0.001 sec, guarantee class: guaranteed, cost: $8 >

4.3. Transport System

With respect to the focused case study, a multi-media transport service is assumed. In order to define a transport service meeting our requirements, the following related work was studied: XTP (express transport protocol) [19], Tenet approach [11,23,24], RTP (real-time transport protocol) [30], OSI 95 [8], BERKOM Multimedia Transport service [4], Heidelberg Transport System [10], and ST-II [32]. Based on this study, a multi-media transport service, MMTS, is suggested. With respect to our case study, MMTS is defined by the following characteristics:

connection-oriented service type

The connection-oriented service type is motivated by the type of application. Multimedia documents require a relatively long term connection. Furthermore, resource reservation facilities are focused which also require a connection-oriented service. A fast-connection mode is not considered because the time for establishing the connection compared with the duration of the connection seems to be very short and so the problem of a fast connection is minor. Also the idea a fast connection is contradictory to the envisaged careful QoS negotiation.

unidirectional point-to-point transmission

The case study requires only a transmission of multimedia data from the server to the client. Control message exchanged between client and server in both direction may not be transmitted within the multimedia connection. Higher facilities such as remote procedure calls or traditionally transport protocol, e.g. TCP/IP, can also be used. Especially remote procedure calls fit better into the ODP architecture providing an object operational interfaces and stream (for isochronous data) interfaces.

transport service data units (TSDU) oriented

The orientation on transport service data units rather then bytes is determined by the character of multimedia data, in particular the frame-structure of digitalized audio and video. Also compressed forms maintain the frame-structure.

quality of service parameters

Resulting from the survey of related work and our own ad hoc case studies, the quality of service parameters shown in Table 3 have been identified.

The first four QoS parameters are used in the majority of comparable approaches. The guarantee parameter the parameters for throughput, delay, and jitter, namely whether they are satisfied by *best effort* or *guaranteed*. The transport service provider will charge its users. This is expressed by the cost parameter. The charge depends on the primary QoS parameters but also on other parameters as daytime, day of the week, etc. The function to calculate the cost is determined by the transport service provider. The reliability parameter determines whether the service is reliable or not. In the unreliable case, a quantitative characterization of the error-rate, e.g. as the number of lost or corrupted packages per second, is given. The considered parameters are summarized in Table 4.

Table 4.
QoS parameters at transport-service-interface.

parameter type	possible values
TSDU-maximum-size	integer (bytes)
throughput	integer (TSDUs/seconds)
delay	real (seconds)
jitter	real (seconds)
guarantee	{ best-effort, guaranteed }
cost	integer ($)
reliability	{ reliable, error-rate }

5. On negotiation protocols

The aim of a negotiation protocols is to determine all parameters in the QoS tuples of all the involved objects according to the QoS architecture, proposed in the previous section. Within the negotiation we identified the following three tasks (see also Figure 4):

- a 3-party QoS negotiation protocol between the client, server, and a multimedia transport service,
- local QoS negotiation at the server and the client, and
- renegotiation.

5.1. Three party QoS negotiation

A negotiation protocol has to include the initial negotiation (i.e. before a connection is established between client and server) and the renegotiation (i.e during a the lifetime of an already established connection). However from our point of view, i.e. above the transport layer, there is no difference in the principles of negotiation and renegotiation.

Figure 6 shows one variant of a QoS negotiation protocol, Figure 7 another. These considerations abstract from the related actions, e.g. querying. In the first case, the negotiation-agent is located at the client side, in the second variant the negotiation-agent is distributed to the client and the servers side. Other variants, e.g. changing the order of the actions in variant I and II or the negotiation-agent locating with a third party object, are possible but considered of minor importance.

The variant I protocol has the following phases:

(1) The client asks for the QoS parameters of a particular multimedia object.

(2) The server provides a set QoS tuples. Multiple QoS tuples occur when the multimedia object is available in different formats or the server provides tools to transform the format.

(3) The client negotiates locally the server's offer with the constraints from its operating system and the wishes of the user. The result of this local QoS negotiation has to be translated into a form corresponding to the MMTS QoS interface.

(4) The client requests from the MMTS a connection with the negotiated QoS parameters.

(5) The MMTS confirms or refuses. (An intelligent MMTS could report possible connections with decreased QoS parameters.)

The variant II of protocol has the following phases:

(1) and (2)
are as in variant I.

(3) The client negotiates locally the server's offer with the constraints from its operating system and the wishes of the user.

(4) The result of this local QoS negotiation is sent to the server.

(5) Translation of the requested QoS parameters.

(6) and (7)
correspond to the steps (4) and (5) of variant I, respectively.

The advantages of variant I are the complete control by the client side and reduced communications. The characteristics of network protocols with resource reservation facilities, like ST-II [32] (the resource reservation goes from the source to the sink), and the unidirectional data flow, however, are better supported by variant II.

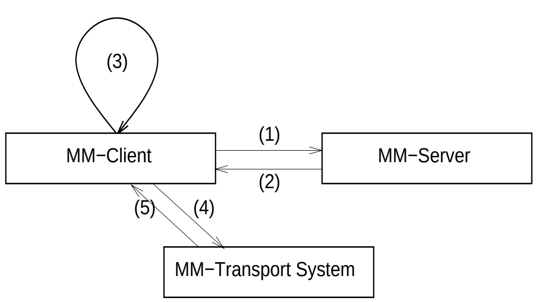

Figure 6. Variant I of QoS negotiation protocols.

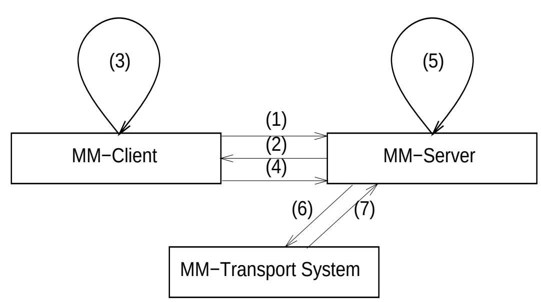

Figure 7. Variant II of QoS negotiation protocols.

5.2. Local QoS negotiation

The local negotiation includes also mechanisms for comparison and translation of QoS parameters. This is also called QoS broker [20,21]. The difficulty seems to be in the definition of algorithms for the translation. For some parameters the situation is clear, e.g. from a certain video or audio format combined with a determined frame rate a throughput requirement for the transport system can be calculated. However, especially the delay parameter which is in presentational applications very much less important than in conversational ones, depends on certain buffer mechanisms which can be used to balance the other parameters, such as jitter or errors. Experiments to determine translation functions for different transport systems are currently under study.

5.3. Renegotiation

Renegotiation do not require a new protocol. With respect to the client and the server, it is only yet another negotiation and the transport service is assumed to hide the renegotiation mechanisms.

A renegotiation can be caused by the following reasons:

- a user's demand for new QoS parameters,
- a monitors report that the transport service does not satisfy the agreed QoS parameters, or
- interrupt of one of the involved components, e.g. the transport system, because it can not satisfy the agreed QoS parameters.

6. Conclusions

We presented an approach to a functional architecture for distributed multimedia presentational applications. In our approach we introduced the concept of shells around a core, the multimedia database. We identified two shells on the user-interface level and the function level. This approach was applied to a multimedia news-on-demand service.

Furthermore, we developed a computational architecture for distributed multimedia presentational applications within the framework of the Reference Model on Open Distributed Processing. We characterized the qualities which distinguish distributed multimedia systems from (general) distributed systems in terms of QoS parameters. Accordingly, the development of the computational architecture was QoS driven. We specified a computational model of the multimedia news-on-demand service within this architecture. It should be mentioned that we developed also a formal approach, i.e. using formal description techniques, to a computational architecture for distributed multimedia systems which is presented in [33].

While other approaches consider QoS parameters specifically for communication (for the transport layer and below), we considered QoS parameters in a more general context. Our approach identifies QoS parameters in all components of the application, the producer, the consumer, and the network (transport layer). We introduced the concepts of the QoS interface through which objects can negotiate their QoS parameters. For the QoS negotiation itself, we investigated various variants of a negotiation protocol.

Currently the implementation of a prototype of the multimedia news-on-demand service is under way. The implementation environment consists of workstations connected by a local ATM switch. This includes also the implementation of an QoS demonstrator as outlined in Section 4.1. Furthermore, extended considerations of QoS negotiation and corresponding protocols for other types of distributed multimedia applications are foreseen. Currently, investigations are under way on a conversational application, the joint viewing and tele-operation service (JVTOS) [13].

Acknowledgements

We would like to thank all who have contributed to this article, in particular our co-investigators in the CITR project, the participants of the seminar on multimedia and high-speed networking held in Université de Montréal during the winter semester 1994, Klara Nahrstedt (University of Pennsylvania) and Ilka Miloucheva (Technische Universität Berlin) for fruitful discussions and comments. Special thanks to Alain Bibal, Thibault Burdin de Saint Martin and Quoc Vu for discussion, comments and contributing to the implementation of the current version of the prototype including the QoS demonstrator and a QoS negotiation protocol.

References

1. M. Altenhofen, et al., "The BERKOM Multimedia Collaboration Service" in *Proceedings of the A - CM Multimedia 93,* ed. P. Venkat Rangan, pp. 457-464, ACM Press, Anaheim (1993).

2. P. B. Berra, F. Golshani, R. Mehrotra, and O. R. L. Sheng, "Introduction Multimedia Information Systems," *IEEE Transactions on Knowledge and Data Engineering,* 5, 4 (Aug. 1993).

3. G. Blair, G. Coulson, and N. Davies, "Communications and Distributed Systems Support for Multimedia Applications," *submitted to a special issue on multimedia information systems of 'Information and Software',* Butterworth-Heinemann.

4. S. Böcking, et al., "The BERKOM MulitMedia Transport System" in *Proceedings of the 1st International Conference on ODP,* pp. 385-391, Berlin (Sept. 1993).

5. A. Campbell, G. Coulson, F. García, D. Hutchinson, and H. Leopold, "Integrated Quality of Service for Multimedia Communications" in *IEEE INFOCOM'93,* pp. 732-739, 1993.

6. C. Chapedelaine, R. Descout, and P. Billon, "Interface Design Issues in the MEDIA-LOG Project" in *Procedings of CASCON'93, Volume II,* ed. A. Gawman, W. M. Gentleman, E. Kidd, P. Larson, and J. Slonim, pp. 707-714, Toronto (1993).

7. S. Christodoulakis, M. Theodoridou, F. Ho, M. Papa, and A. Patria, "Multimedia Document Presentation, Information Extraction, and Document Formation in MINOS : a Model and a System," *ACM Transactions on Office Information Systems,* 4, 4 (Oct. 1986).

8. A. Danthine, "OSI 95. High Performance Protocol with Multimedia Support on HSLANs and B-ISDN" in *3th Joint European Networking Conference,* Insbruck (1992).

9. L. Delgrossi, R. G. Herrtwich, and F. O. Hoffmann, "An Implementation of ST-II for the Heidelberg Transport System" in *IEEE Globcom'92,* Orlando (1992).

10. L. Delgrossi, et al., "Media Scaling for Audiovisual Communication with the Heidelberg Transport System" in *Proceedings of the ACM Multimedia 93,* ed. P. Venkat Rangan, pp. 99-104, ACM Press, Anaheim (1993).

11. D. Ferrari, A. Banerjea, and H. Zhang, "Network Support For Multimedia," TR-92-072, The International Computer Science Institute, Berkley, CA (1992).

12. Homsy G., Govindan R., and Anderson D. P., *Implementation Issues for a Network Continuous-media I/O Server,* The International Computer Science Institute, Berkeley (1990).

13. T. Gutekunst, T. Schmidt, G. Schule, J. Schweitzer, and M. Weber, “A Distributed Multimedia Joint Viewing and Tele-Operation Service for Heterogeneous Workstation Environments” in *Proceedings of International Workshop on Distributed Multimedia Systems,* Stuttgart (1993).

14. G. Kalkbrenner, T. Pirkmayer, A. van Dornik, and P. Hofmann, “Quality of Service (QoS) in Distributed Hypermedia-Systems” in *Procceedings of RPODP’94 - Second International Workshop on Principles of Document Processing,* Darmstadt (April 1994).

15. B. Kerherve, G. v. Bochmann, R. Dssouli, J. Gecsei, A. Hafid, and A. Vogel, “Distributed Multimedia Presentational Applications Requirements,” Technical Report #893, Université de Montréal (1993).

16. L. Lamont, L. Li, and N. D. Georganas, “Synchronization Architecture and Protocols for a Multimedia News Service Application,” Technical Report, University of Ottawa (1994). accepted for Broadband Islands’94 Conference.

17. C. Meghini, F. Rabitti, and C. Thanos, “Conceptual Modeling of Multimedia Documents,” *IEEE Computer* (Oct. 1991).

18. G. Miller, G. Baber, and M. Gilliland, “News On-Demand for Multimedia Networks” in *Proceedings of the A - CM Multimedia 93,* ed. P. Venkat Rangan, pp. 383-392, ACM Press, Anaheim (1993).

19. I. Miloucheva, “XTP and ST-II Protocol Facilities for Providing the QoS Parameters of Connection-Mode Transport Services,” Research Note TUB-PRZ-W-1029, Technical University Berlin, Berlin (1992).

20. K. Nahrstedt and J. M. Smith, “Application-Driven Approach to Networked Multimedia Systems” in *Proceedings of the 18th Conference on Local Computer Networks* (1993).

21. K. Nahrstedt and J. M. Smith, “Revision of QoS at the Application/Network Interface,” Technical Report MS-CIS-93-94, University of Pennsylvania (1993).

22. S. R. Newcomb, N. A. Kipp, and V. T. Newcomb, “The HyTime Hypermedia/Time-based Document Structuring Language,” *Communications of the ACM,* 34 (1991).

23. C. J. Parris, G. Ventre, and H. Zhang, “Graceful Adaption of Guaranteed Performance Service Connections,” TR-93-011, The International Computer Science Institute, Berkeley (1993).

24. C. J. Parris and D. Ferrari, *A Dynamic Connection Management Scheme for Guaranteed Performance Services in Packet-Switching Integrated Services,* The International Computer Science Institute, Berkeley (1993).

25. R. Price, “MHEG: An Introduction to the Future International Standard for Hypermedia Object Interchange” in *Proceedings of the First ACM International Conference on Multimedia,* ed. P. Venkat Rangan, pp. 121-128, ACM Press, Anaheim (1993).

26. P. V. Rangan, “Video conferencing, file storage, and management in multimedia computer systems,” *Computer Network and ISDN Systems,* 25, pp. 901-919 (1993).

27. K. Raymond, “The Reference Model of Open Distributed Processing: a Tutorial” in *Proceedings of the 1st International Conference on ODP,* ed. J. de Meer, B. Mahr, and O. Spaniol, pp. 3-14, Berlin (1993).

28. L. A. Rowe and R. R. Larson, “A Video-on-Demand System,” Project proposel, University of California at Berkeley (1993).

29. L. A. Rowe and B. C. Smith, “Performance of a Software MPEG Video Decoder” in *Proceedings of the First ACM International Conference on Multimedia,* ed. P. Venkat Rangan, pp. 75-82, ACM Press, Anaheim (1993).

30. H. Schulzrinne, “RTP: The real-time transport protocol” in *MCNC 2nd Packet Video Workshop,* 2, Research triangle park (Dec. 1992).

31. R. Steinmetz and C. Engler, “Human Perception of Media Synchronization,” Technical Report 43.9310, IBM Europeen Networking Center, Heidelberg (1993).

32. C. Topolcic, “Experimental Internet Stream Protocol, Version 2 (ST-II),” Internet RFC 1190 (1990).

33. A. Vogel, “A Formal Approach to an Architecture for Open Distributed Processing,” Technical Report #902, Université de Montréal (1994).

34. J. Wong, et al., “CITR Major Project ‘Broadband Services’,” Project Description, Canadian Institute for Telecommunication Research.