# Relatório Técnico

Núcleo de Computação Eletrônica

# Design of an Integrated Enviroment for Adaptive Multimedia Document Presentation Through Real Time Monitoring

Eduardo Carneiro da Cunha
Luiz Fernando Rust da Costa Carmo
Luci Pirmez

NCE - 30/99

Universidade Federal do Rio de Janeiro

# Design of an Integrated Environment for Adaptive Multimedia Document Presentation Through Real Time Monitoring

Eduardo Carneiro da Cunha,
Luiz Fernando Rust da Costa Carmo,
and Luci Pirmez
E-mail: educc, rust, luci@nce.ufrj.br

Núcleo de Computação Eletrônica - UFRJ
P.O.Box 2324, 20001-970
Rio de Janeiro – RJ – Brazil

**Abstract.** The retrieval of multimedia objects is influenced by factor such as throughput and maximum delay offered by the network, and has to be carried out in accordance with the specification of object relationships. Many current network architectures address QoS from a provider's point of view and analyze network performance, failing to comprehensively address the quality needs of applications. The work presented in this paper concerns the development of an integrated environment for creation and retrieval of multimedia documents, that intends to preserve the coherence between the different media, even when the process is confronted with a temporary lack of communication resources. This environment implements a communication system that address QoS from the application's point of view and can help in handling variations in network resources availability through a real-time monitoring over these object relationships.

## 1 Introduction

Multimedia systems are computer systems that manipulate, in an integrated way, several types of media information. Frequently, multimedia systems are distributed; that is, its components are located at different processing nodes in a local or wide LAN. Quality of service in this context can be intuitively defined as a measure of how satisfied the user is with regard to a service rendered by a multimedia distributed system (DMS). Although the notion of QoS is intuitive, a series of measurable parameters can be established to define such concept objectively. These parameters are divided in two levels: system and user.

Most of the existing DMS architectures treats the quality of service from the system's point of view, that is, from the provider's point of view, and they use effective monitoring politics and resource management to provide quality of service support [1]. However, due to the heterogeneous nature and varying capabilities of today's end-systems and global network infrastructures, conventional resource reservation and admission techniques cannot guarantee QoS without considerable over-booking and inefficient resource utilization. Furthermore, these architectures fail to raise the QoS notion up to the user level, causing the QoS specification and management to be made through the system level parameters.

To address this problem, and avoid any adverse impact to the end-user, distributed applications and their infrastructures need to be adaptive. This means that either applications must tolerate fluctuations in resource availability or that the supporting infrastructure can itself mould to the dynamically changing requirements of the applications.

The work presented in this paper is originated from a research project on distance teaching that is being developed at NCE/UFRJ. Basically, this project, named ServiMedia [6], deals with multimedia documents authoring and storage, and with the network infrastructure for the remote retrieval of those documents. The basic subject is how accommodate in the same client/server, distance teaching environment, users with substantial differences on communication resources availability. The same document, retrieved by different users, can be interpreted in a form completely different, in agreement with the quality of the presentation that is noticed by the user. The quality of the presentation, in its turn, is directly related to the readiness of resources found in the network along the path that reaches the user, since some of the multimedia flows can be lost or partially damaged during the process of document retrieval.

In this work, we investigate a strategy for composition, storage and retrieval of multimedia documents that allows us to generate different formats of presentations at the user's site, starting from just one multimedia document specification at the document server. These presentation formats are adapted from the original specification in agreement with the network resources availability verified on the routes to the respective users. The key point in this strategy is that each generated format must preserve the semantic properties of the media originally specified by the document's author. This means that the multimedia document should maintain its semantics even when it is adapted and suffers some degradation in relation to the original document specification.

This paper is structured as follows: the section 2 presents the authoring strategy developed in this project and shows how to implement the strategy through extensions to the synchronized multimedia integration language (SMIL). The section 3 presents the ServiMedia Architecture and how the adaptive retrieval mechanism works. Still in the section 3, we describe the real-time stream relationship monitoring and we present an application example. Finally, the section 4 relates some conclusions of this work.

## 2 Composing a Multimedia Document

Current authoring systems accomplish the specification of multimedia documents based on three fundamental aspects: the logical structuring of the presentation, the establishment of spatial positioning and temporal relationships between the multimedia objects.

It is important to highlight that the concern about the maintenance of presentation coherence and semantic, associated to its QoS degradation control, and the inclusion of this facility in the authoring phase, is not explored by the current DMS architectures, which are most of the time just focused on the temporal issues. In order to provide the specification of semantic relationships between the multimedia objects, we use a modified version of the causal synchronization model presented in [3]. Thus, we've defined an authoring strategy that combines:

1) mechanisms of logical structuring of the presentations: the logical structuring worries about establishing abstraction mechanisms, intending to obtain a wide and structured view of the presentation. We use a logical structure that is based on the concept of groups of clips (parallel and sequencial), where the clips are the media objects that compose the document. These groups are represented in a hierarchical tree similar to a tree of directories.
2) a model of spatial synchronization based on the definition of playback areas.
3) a model of temporal synchronization based on timelines: this imposes rules on how the objects can be linked to each other. Several models have been proposed in the literature, obeying, as possible, to some basic requirements pointed in [2]. A flexible temporal specification is obtained through the establishment of margins of tolerance for the beginning of the presentation of each object (fig.1). The advantage of the use of flexible temporal specifications is that it facilitates the use of relaxation and acceleration techniques of the presentations with synchronization purposes, aiming in the derivation of a schedule of the presentation as discussed in [8].
4) a model of causal synchronization based on conditional dependencies between the involved objects.

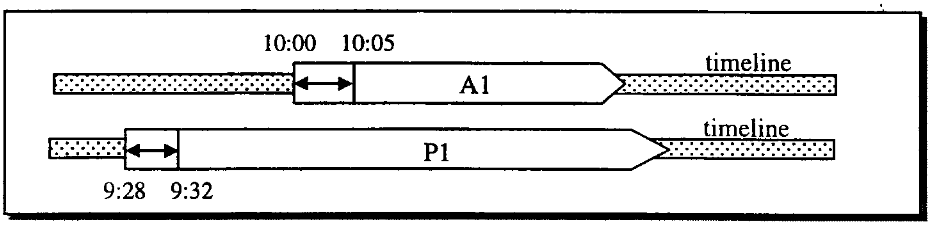


**Fig. 1.** Flexible temporal specification. Establishiment of a temporal interval in the beginning of an object indicates that its presentation can start at any instant within this interval.

## 2.1 Expressing Conditional Dependencies

Conditional dependencies have been proposed [4, 5] as a way of taking advantage of the knowledge on semantic relationships between different stream's objects. Conditional dependencies are causal relations associated with a stream's object aiming to express delivery constraints of that object, relative to the delivery of other objects belonging to either the same stream (intra-stream conditional dependencies) or distinct ones within the same bundle (inter-stream conditional dependencies).

To address the problem of different communication resource availability the author is allowed to specify the importance of each object that he inserts in the document and to delineate the QoS requirements and conditions that should be respected in order to preserve the consistency of the document.

Associated to each object, there is a *QoS descriptor* that defines the possible quality degradation. In the QoS descriptor the quality degradation is specified in terms of a range, within which the object can be manipulated to handle possible variations on the network resources.

Thus, the author can define whether an object is essential or just qualitative and also establish causal relationships between the qualitative ones. These relationships are specified through links that interconnect the qualitative objects forming a net of causality that describes the coherence wanted for the document [7].

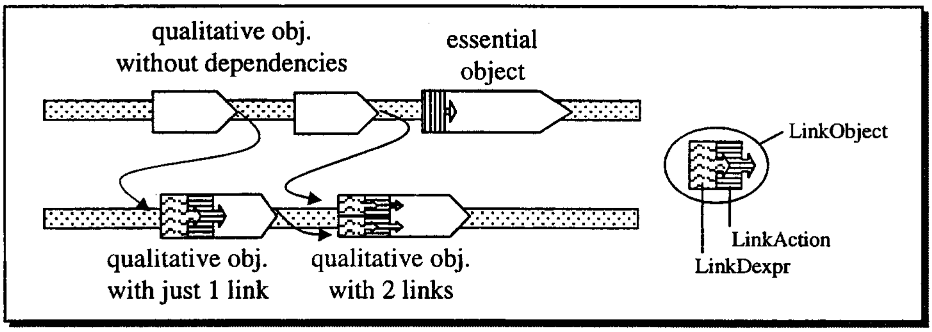


**Fig. 2.** Links representation. The Link objects are coupled to the respective qualitative objects.

In [7], we define a ***Link object*** that consists of a ***LinkAction*** and a ***LinkDexpr***. LinkDexpr is a Boolean expression (dependency expression) that characterizes the causal relations through conditions associated to other qualitative objects. Two link types were defined: ***startlink*** and ***stoplink***. Observe that the links are fired by temporal synchronization. Both the temporal and causal synchronization must be respected.

### 2.2 Implementing the authoring strategy

In relation to the multimedia presentation authoring, there is not a consent or standard widely accepted for the specification of multimedia documents that should be retrieval or presented through remote servers. The Synchronized Multimedia Integration Language (SMIL) [9] allows the integration of an independent multimedia group of objects in a synchronized multimedia presentation through a textual specification, with tags and very similar to HTML. In particular, SMIL is a format of multimedia data description for authoring tools and players.

SMIL introduces many valuable ideas that are similar to our authoring strategy and that can be used by the ServiMedia environment. This way, we've decided to adopt the language SMIL as reference for our ServiMedia authoring system. However, certain characteristics specific of our authoring strategy are not considered by SMIL 1.0. For example, the flexible temporal specification and the causal relationships through links are not considered. We have been working on this issue in order to describe the extensions to SMIL so as to implement our authoring strategy [7].

To allow a flexible temporal specification we have created the ***can-begin*** attribute that defines an interval of tolerance for the beginning of the presentation of any clip. To specify the links that describe the causal relationships between the clips of a document, we have created two attributes (one for each link type): ***startlink*** e ***stoplink***.

```
<par>
   <audio id="A1" src="..." can-begin="5s"
                  startlink="(P1:started)"/>
   <img id="P1" src="..." can-begin="4s"
                  stoplink="(A1:concluded)"/>
</par>
```

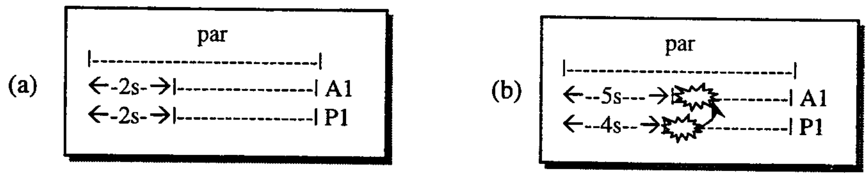


**Fig. 3.** (a) P1 and A1 begin synchronized at 2s and they finish synchronized when A1 finish; (b) it is not possible to begin P1 within the tolerance, P1 and A1 are discarded then.

The SMIL *switch* tag carries out the choice based on static variables that are configured or stored in the clients presentation tools (players). Using the causal relationships, that is, the ***startlink*** and ***stoplink*** attributes, we are testing dynamic variables that change their states during the presentation. Actually, we can even associate a causal relationship to the switch group, as shown in [7], obtaining a dynamic adaptation of the presentation. It depends only on the author of the document to specify the possible variations in the presentation formats.

## 2.3 The Authoring System Prototype

In this section we present a general view of the authoring tool developed in this project. It uses the language SMIL increased with the extensions that have been created and presented in the previous section. This tool was initially developed for the platform Windows98®.

Figure 4 illustrates one of the authoring tool interfaces. Through this interface, the author specifies the layout defining the playback areas where the clips must be presented. It is possible to create new areas, to move, resize and define its background colors. For each new area a new timeline is added where the objects (clips) can be placed. One timeline for the sound track of the presentation is always present.

In another interface the author specifies the logical structure of the presentation through a hierarchical tree. In this interface it is possible to create new groups (*par,* *seq* and *switch*), new clips, to organize them, and to define all its attributes and properties. All modifications done in the presentation structure, in the groups and its properties, are reflected in the visualization of the timeline of each area of the layout.

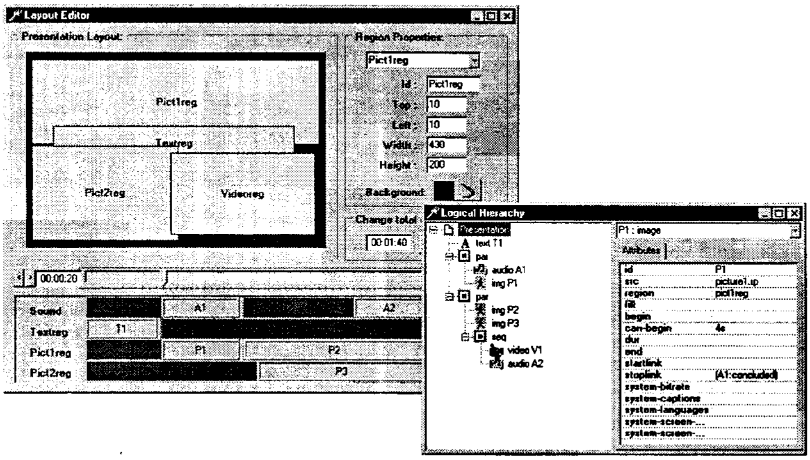


**Fig. 4.** Layout editor interface and presentation editor interface

## 3 Retrieving Multimedia Documents

A protocol for multimedia document retrieval control has been designed by the IETF recently. The real time stream protocol (RTSP) [13] is a client/server control protocol for multimedia presentation. RTSP was developed to deal with the needs of an efficient distribution of multimedia streams within IP networks. RTSP establishes and controls either a single or several time-synchronized streams of continuos media such as audio and video. In other words, RTSP acts as a network remote control for multimedia servers, and consists basically of request and response messages. The requests are issued sequentially on different connections.

### 3.1 Real-time delivery and monitoring

Currently, there is an increasing demand for real-time applications that transfer continuous media. These new multimedia application, such as video conferencing and media-on-demand, impose new QoS requirements in terms of delay, error rate, jitter and throughput. The real-time applications must deliver data within an expected time frame and thus they cannot depend on TCP in the real-time transmission. Hence, in order to meet these new requirements, the Real-Time Transport Protocol (RTP) has been designed and it is used nowadays for real-time continuous media transmission over the Internet.

The real-time transport protocol (RTP) is both a IETF Proposed Standard [10] and a ITU Standard (H.225.0). RTP provides end-to-end delivery services for data with real-time characteristics, such as interactive audio and video. Those services include payload type identification, sequence numbering, timestamping and delivery monitoring. However, RTP itself does not provide any mechanism to ensure timely delivery or provide other QoS guarantees, but relies on lower-layer services to do so.

The data transport is augmented by a control protocol (RTCP) to allow monitoring of the data and to provide minimal control and identification functionality. The RTP control protocol (RTCP) is based on the periodic transmission of control packets to all participants in the session, using the same distribution mechanism as the data packets. RTCP provides feedback on the quality of data delivery. The feedback may be directly useful for control of adaptive multimedia applications. This feedback function is performed by the RTCP sender and receiver reports. The RTP functions require that all participants send RTCP packets, therefore the rate must be controlled in order for RTP to scale up to a large number of participants. The RTCP scales well for a small multicast group but a scalability problem arises when it comes to a group of thousands of users. Some of these problems are addressed in [11, 12].

Cumulative counts are used in both sender information and receiver report blocks so that differences may be calculated between any two reports to make measurements over both short and long time periods, and to provide resilience against the loss of a report. The difference between the last two reports received can be used to estimate the recent quality of the distribution. For example, we can calculate the number of packets lost during an interval, the number of packets expected during an interval, the packet loss fraction over an interval, the loss rate per second, the apparent throughput available to the receiver and the interarrival jitter. Packet loss tracks persistent congestion while the jitter measure tracks transient congestion.

### 3.2 Resource reservation

Our communication architecture allows a resource reservation for the transmission of essential information. That is, for the traffic of essential information, a resource reservation protocol that communicates with a QoS routing algorithm should be used so as to select the best route capable to absorb the traffic and the resource reservation [16]. On the other hand, the transmission of qualitative information is accomplished without warranties through the routes statistically more favorable to the success of the transmission. This means that for the traffic of qualitative information the reservation is not carried out but the QoS routing algorithm is still used to choose the most favorable routes. That guarantees, at least, the retrieval and presentation will be in conformity with the basic QoS requirements specified by the author. The routing algorithm is based on the combination of two metrics: the bandwidth of each communication channel and the end-to-end delay variation generated by the route [17, 18].

RSVP [14] has been designed to support resource reservation in the Internet. However, it has two major problems: complexity and scalability. The former results in heavy message processing overhead at end-systems and routers. The latter implies that the amount of bandwidth consumed by refresh messages and the storage space that is needed to support a large number of flows at a router are too large, mainly in a backbone environment. Our next direction in the project is to investigate resource reservation associated with QoS routing. We are studying the RSVP complexity issues within the Integrated Services Model and the recent Differentiated Services Model. We intend to define a scheme that will use effectively the real-time protocols (RTP/RTCP and RTSP) not only to delivery continuous media but also to provide resource reservation and QoS-routing while addressing the complexity and scalability problems mentioned in [11, 12, 15, 19].

### 3.3 Adaptive retrieval

QoS adaptation is the process of maintenance control, facilitated through alterations to either the balance and distribution of resources or to the application's level of service, on short time scales. Adaptation processes often occurs as a result of QoS notifications, usually emitted from QoS monitoring mechanisms, which indicate a change in the observed service affected through the availability of some element of the end-to-end resources. Notifications may indicate a imminent lack of resources and hence reduction in service quality (QoS degradation) or a failure to maintain service quality through a complete loss of resources (QoS failure). By QoS failure we mean either a complete route failure or an impossibility in keeping service quality inside the QoS requirement range provided by the QoS descriptor.

We can find in the literature a diversity of QoS adaptation mechanisms, such as QoS filters, sender rate adaptation, layered multicast, etc. However, all of them are concerned only with the traffic parameters and with the direct user perception of isolated media objects. The whole presentation, with all its media streams and their semantic relationships, is not treated as a complete documentation that has significance just due to the combination of several information objects. In the ServiMedia, starting from the document generated by the authoring system, a global QoS scenario is created. This scenario consists of the QoS requirements and conditional dependencies of the streams. An mechanism of dynamic path allocation and adaptive retrieval uses this scenario. This mechanism is implemented through two complementary modules: QDM (QoS descriptor monitor) and SRM (Stream relationship monitor).

This mechanism operates based on a real-time monitoring over the QoS scenario of qualitative streams, so as to achieve three main goals: 1) manage the delivery of qualitative streams, 2) verify whether the QoS requirements are being respected and adapt the QoS to fluctuations in the network resources, and 3) adapt the document structure based on the causal relations between the qualitative streams. Through this mechanism, we can release routes and resources that are being used for streams whose delivery, from the application point of view, became useless. We can also start delivering streams with less QoS requirements to replace other streams whose QoS requirements could not be respected.

An increasing number of current multicast applications prefer to use receiver-based adaptation schemes instead of sender-based adaptation schemes to adapt to congestion in the network. In sender-based adaptation, when congestion occurs, the sender decrease its rate of data transmission to suit the receiver with lowest capabilities.

Receiver-based adaptive applications have the advantage of accommodating to the heterogeneous capabilities and conflicting bandwidth requirements of different receivers in the same multicast group [20]. Besides, the real-time protocols represent a new style of protocol following the principles of application level framing (ALF) and integrated layer processing proposed in [21] and used in [22]. That is, these protocols intend to be malleable to provide the information required by a particular application and will often be integrated into the application processing rather than being implemented as a separate layer. Thus, the QDM and the SRM were implemented at the application level of the receiver side (fig.5).

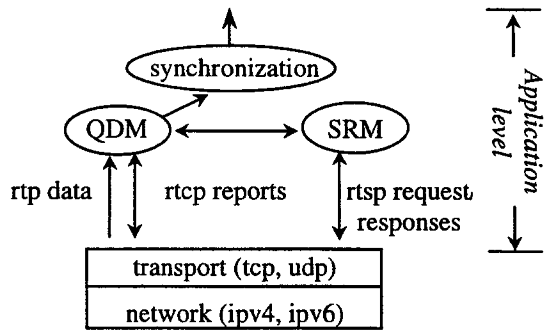


**Fig. 5.** Receiver-based adaptive structure. The QDM is responsible for monitoring the QoS feedback provided by the RTCP messages in order to preserve the QoS requirements as well as, when the QoS adaptation exceeds the QoS descriptor limits, to signal the SRM to adapt the document so as to preserve the causal relationships between the streams, in other words, to preserve the document consistency.

### 3.4 Overall Operation

An RTSP URL may identify each presentation and media stream. The overall presentation and the properties of the media the presentation is made up of are defined by a presentation description file. The presentation description file may be obtained by the client using HTTP or other means and may not necessarily be stored on the same server as the media streams.

The presentation description file contains a description of the media stream making up the presentation, including their encodings, language, QoS requirements, and other parameters that enable the receiver (client) to choose the most appropriate combination of media. In this presentation description, each media stream that is individually controllable by RTSP is identified by an RTSP URL, which points to the media server handling that particular media stream and names the stream stored on that server. Several media stream can be located on different server for load balancing purposes [23]. Following, we identified the components that participate in the architecture illustrated in the figure 6:

***Document Server*** is the entity that receives the retrieval requests from the clients. The Document Server stores the specification of the multimedia document. An agent located in the document server receives the retrieval requests and, based on the information obtained from a media distribution information base (DIB), it find out how the information is replicated through the serves and verifies which media servers should be signaled to begin the transmission of the streams belonging to the document requested by the client. Then, it composes the presentation description file automatically inserting a list of URLs for each stream that points to different media copies.

***Distribution Information Base (DIB)*** corresponds to a MIB extension for management through the SMNP protocol. Here the information about the media copies location over the several media servers are stored, as well as information on the topology and on the available network resources. These informations contribute to a choice of the media servers that reach a best load sharing between media servers.

***Media Servers*** store the digital media information. The Media Servers may exist in any number and several media streams can be split across servers for load sharing. They receive the RTSP requests from the clients for the transmission of multimedia streams creating new RTSP sessions.

***ServiMedia Clients*** are responsible for giving the departure in the processing of a document when requesting its retrieval to the Document Server. After receiving the presentation description, the presentation system starts the communication with the media servers. Afterwards, the QoS monitoring mechanisms (QDM and SRM) starts monitoring the communication to preserve the QoS requirements as well as the causal relationships specified by the author in the presentation description.

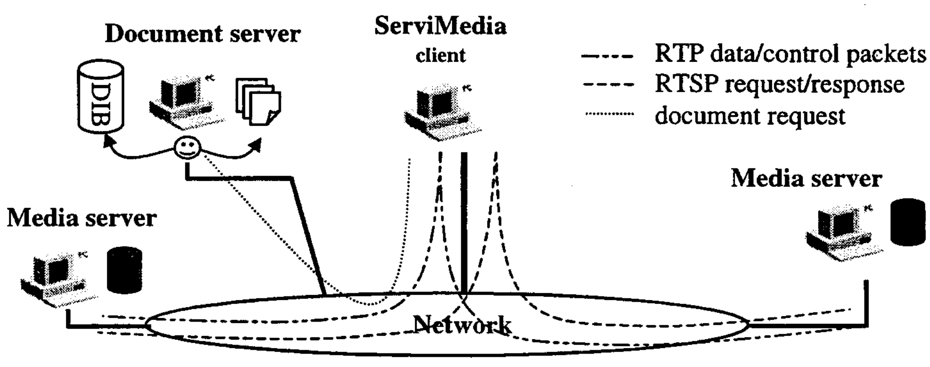


**Fig. 6.** Architecture of the ServiMedia Environment

In addition to the recent tendency in receiver-based adaptation scheme, there are at least three reasons for implementing the SRM as an integral part of the presentation system at the client site:

1. The presentation system possesses the information (contained on the presentation description) about the stream interdependencies. This way, it knows which causal relationships are to be monitored;
2. Once the presentation description, with the addresses of the media servers supplied by the document server, has been obtained, the RTSP sessions are

established between the client and the servers. Thus, the client can monitor the transfer of streams through the RTCP messages and send RTSP requests to stop, pause or start streams. If a client can start a stream, it must be able to stop and control the stream. Server should not start streaming to clients in such a way that clients cannot stop the stream.

3. Deciding that the monitoring should be made by the presentation system, we are adopting a decentralized control configuration where each client is responsible for monitoring the sessions (streams) that were established between it and the servers. One could think about a centralized configuration, where the SRM module could be implemented for example, in the document server. In spite of the document server also possessing the description of the stream interdependencies, it would have to participate in all RTSP sessions established between clients and servers for each one of the presentations. Besides, for the case of a multicast session with receiver diversity, it is much more suitable that the client makes the decision of abandoning or not the session while other clients stay connected receiving the corresponding flow.

### 3.5 Application Example

In this section, a simple example in the area of distance and interactive training is described. The application is assumed to be distributed over three nodes, the document server and two media servers. The training application comprises the following two parts: an introduction (the servers send information to the student in order to present the authors and copyrights, and give two option of training subject) and a training part (the servers send information so as to present the training material).

In the *introduction*, an opening video stream (IV1) presents the company providing the training and the credits, the authors are also presented. In parallel with the video there is a background sound track (IA1). An optional picture (IP1) is presented if the video cannot be presented with the specified QoS requirement. Following, a picture with two icons (IP2), representing the two possible trainings is presented. These two icons link the introduction to other specific presentation that contains the selected training (*training1* or *training2*). When the client request the document via http, the document server, after consulting the DIB, decides to stream IV1 from server 1 and IA1 from server 2.

Let's assume that the student has selected the *training2*. At this point, the process repeats, that is, the client request the document *"training2.smil"* from the document server, who consults the DIB and decides which severs should the streams be retrieved from. In the *training* phase, the document contains a video stream (TV1) that presents the training content, an audio stream (TA1) related to TV1, and a picture stream (TP1) that presents diagrams and other static images. In this phase there is a strong requirement for the delivery of the video stream. In other words, the delivery of only the audio and picture information is regarded as useless from an application point of view. If only the delivery of the audio information cannot be performed, it will be replaced by delivering a text information (TT1) to the student.

Figure 7 summarizes the communication process between client and servers for the training phase of the application. It shows the case where the audio TA1 cannot be delivered from the mediaserver2 within the QoS descriptor limits causing QDM to notify a QoS failure to SRM. SRM, in turn, examines the presentation description and decides to deliver the text TT1 from mediaserver2 to replace TA1. After that, SRM updates QDM with information about the new presentation structure (new streams) so that QDM can continue monitoring the quality of data delivery.

We consider that in a distance teaching environment, where the presentations (training) are most of the time relatively long, the receiver will not be bother by a delay caused by adaptation mechanisms, since we think the significance and consistency of the document are more critical than a initial delay. We assume that the receiver (student) will prefer to watch a coherent presentation with some casual delays than a continuous presentation with an unsatisfactory quality and that does not make sense. This situation comes to emphasized the importance of a flexible temporal specifications mentioned in the section 2.

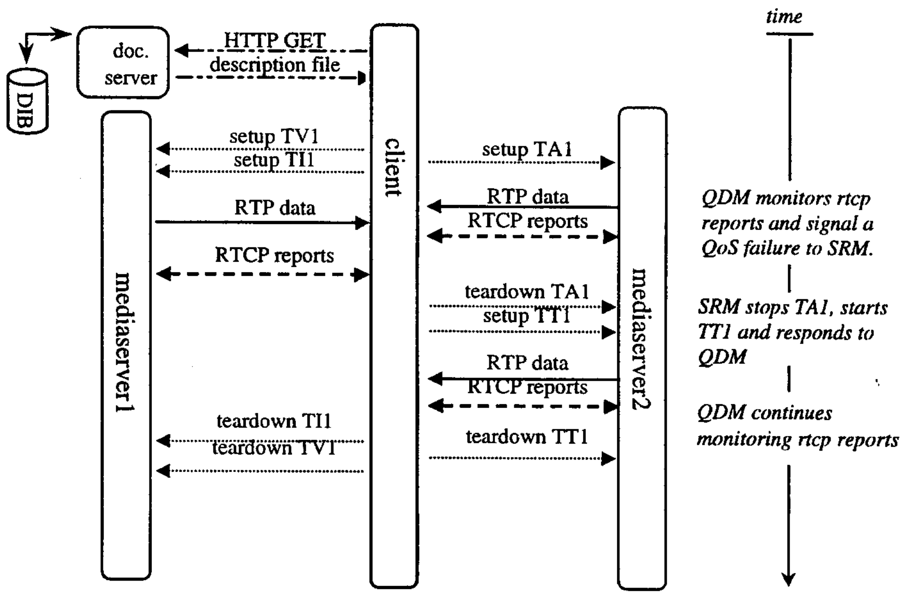


**Fig. 7.** Communication process

## 4 Conclusions

The authoring strategy presented in [7] promotes a strong integration between the systems that compose the ServiMedia environment. The establishment of a net of causal relationships in a document generates subsidies that allow the other systems (communication and presentation) to decide when adapting some information and how this adaptation must be carried in order to preserve the document semantic. That characteristic has been of great value, mainly in an integrated distance teaching environment.

An authoring system was developed based on the Synchronized Multimedia Integration Language (SMIL). A presentation system is under development in order to carry out the QoS monitoring and adaptation through the utilization of the QoS descriptor and the stream relationships. It is used the feedback provided by the RTCP reports and the real-time stream protocol to control the presentation. This real-time monitoring mechanism (QDM + SRM) allows the client to manage the QoS requirements as well as to preserve the causal relationships specified by the author. The whole presentation, with all its media streams and their semantic relationships, is considered as a complete documentation that must preserve its significance, which has been shown to be of great value in our distance teaching environment.

# 6 Reference


1. Aurrecoechea C, Campbell AT, Hauw L, "A survey of QoS architectures", ACM Multimedia Systems 6:138-151, 1998.
2. G. Blakowski, R. Steinmetz, "A Media synchronization Survey: Reference Model, Specification, and Case Studies", IEEE Journal on Selected Areas in Communications, Vol. 14, N. 1, pp. 5-35, January, 1996.
3. J.P. Courtiat, L.F.R.C. Carmo, R.C. de Oliveira, "A General-purpose Multimedia Synchronization Mechanism Based on Causal Relations", IEEE Journal on Selected Areas in Communications, Vol. 14, N. 1, pp. 185-195, January, 1996.
4. J.P. Courtiat, L.F.R.C. Carmo, R.C. de Oliveira, "A new mechanism for achieving inter-stream synchronization in multimedia communication systems", in Proc. IEEE Int. Conf. on Multimedia Comput. Systems, Boston, , pp.173-182, May 1994.
5. J.P. Courtiat, R.C. de Oliveira , L.F.R.C. Carmo, "Toward a new multimedia synchronization mechanism and its formal specification", in Proc. ACM Multimedia'94, San Francisco, pp. 133-140, October 1994.
6. L.F.R.C. Carmo, L.Pirmez, "ServiMedia: An Integrated System for Multimedia Document Creation and Retrieval with Adaptive QoS Control", 2° Franco-Brasilian Symposium on Distributed Computer Systems, Brazil, November, 1997.
7. E. C. da Cunha, L.F.R.C. Carmo, L. Pirmez, "A Multimedia Document Authoring Strategy for Adaptive Retrieval in a Distributed System", 6th International Conference on Distributed Multimedia Systems, University of Aizu-Japan, July 1999.
8. Candan KS, Prabhakaran B, Subrahmanian VS, "Retrieval schedules based on resource availability and flexible presentation specifications", ACM Multimedia Systems 6: 232-250, 1998.
9. Synchronized Multimedia Working Group of the W3 Consortium, "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification", W3C Recommendation, June, 1998.
10. H. Schulzrine, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC1889, January, 1996.
11. J. Rosenberg, H. Schulzrine, "Timer Reconsideration for Enhanced RTP Scalability", draft-ietf-avt-reconsider-00.ps, July 1997.
12. J. Rosenberg, H. Schulzrine, "New Results in RTP Scalability", draft-ietf-avt-byerecon-00.ps, November 1997.
13. H. Schulzrine, A. Rao, R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC2326, April, 1998.
14. R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Spec. RFC2205, September, 1997.

15. P. Pan, H. Schulzrine, "YESSIR: A Simple Reservation Mechanism for the Internet", August 1997
16. E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, "A Framework for QoS-Based Routing in the Internet", RFC2386, Agosto 1998.
17. Z. Wang and J. Crowcroft, "Quality of Service Routing for Supporting Multimedia Applications", IEEE J. Selected Areas Communication, 14 (7), 1996.
18. R. Vogel, R. Herrtwich, W. Kalfa, H. Wittig, and L. Wolf, "QoS-Based Routing of Multimedia Streams in Computer Networks", IEEE J. Selected Areas Communication, 14(7), 1996.
19. R. El-Marakby, D. Hutchison, "A Scalability Scheme for the Real-time Control Protocol", Proceedings of the 8th IFIP International Conference on High Performance Networking (HPN'98), pp.147-162, Vienna, Austria, September 1998.
20. D. Waddington, D. Hutchison, "End-to-End QoS Provisioning through Resource Adaptation", Distributed Multimedia Research Group, Lancaster University, Internal report, "ftp://ftp.comp.lancs.ac.uk/pub/mpg/MPG-98-10.ps.gz" , May 1998.
21. D. Clark and D. Tennenhouse, "Architecture considerations for a new generation of protocols", in ACM SIGCOMM'90, pp. 200-208.
22. I. Chrisment, D. Kaplan, C. Diot, "An ALF Communication Architecture: Desing and Automated Implementation", IEEE Journal on Selected Areas in Communications, Vol.16, Nº.3, pp.332-344, April 1998.
23. D. Pegler, N. Yeadon, D. Hutchison, D. Shepherd, "Incorporating Scalability into Networked Multimedia Storage Systems", technical report, Department of Computing, Lancaster University, 1996.