An Architecture for Interactive Distributed Multimedia Information Services

C. Bouras[1,2] V. Kapoulas[1,2] D. Miras[1] V. Ouzounis[1] P. Spirakis [1,2]

[1] *Computer Engineering and Informatics Dept.*

Univ. of Patras, 265 00 Patras, Greece

[2] *Computer Technology Institute*

262 21 Patras, Greece

E-Mail: bouras@cti.gr

Abstract

In this work, we present an architecture for interactive distributed multimedia information services. The service consists of three basic parts, the clients, the access servers, and the content servers. Clients manage the presentation of the requested multimedia documents, while also ensure the synchronization among various involved media. Special policies for the synchronized presentation of multimedia documents and user interactions management are discussed. Access servers authorize the users for legitimate use of the service, while also provide various searching capabilities Content servers manage the transportation of multimedia documents over a broadband network and the user interactions. The proposed architecture preserves scalability, flexibility and modularity due to the distributed nature of the architecture, while also results in good performance due to the dynamic allocation of resources.

1. Introduction

During the recent years we are experiencing an essential development of the multimedia technologies and a rapid progress in the network infrastructure. Some of the enabling technologies that are responsible for this growth are high speed networking and communication protocol standards, compression standards for images and video, high speed computing and the influence of the WWW. These developments make feasible the delivery of multimedia information among places over the network and contribute to the emergence of new applications that make use of multiple media. Advanced multimedia systems are characterized by the integrated computer-controlled generation, storage, communication, manipulation and presentation of independent time-dependent and time-independent media [7].

A multimedia information system can be considered as a set of functions or services that intend to present to the end user interactive information that invokes a variety of media, such as formatted text, images, slides, graphs, animation, audio, sound, video, etc., in a unified way [3]. These applications are bandwidth and memory intensive, so it becomes very important to develop schemes which efficiently utilize system and network resources while simultaneously provide the best service to the users [1,4,5,6]. To be successful, these applications need to make huge volumes of data easily accessible to their users. To make data handling manageable, the information should be distributed across different sites [6].

As the number of users that desire to use these services increases, new policies for the retrieval of distributed multimedia data and the on-demand delivery of them should be addressed. These policies should support flexible access to the information, security, privacy, accounting, intellectual property rights management, anonymity of the users, accounting mechanisms, and finally, resource allocation and management. Most of the existing trials in the area of distributed multimedia services suffer from limited accessibility, insufficient security, low level of integration and scalability, inefficient resource allocation mechanisms and policies.

In this paper, we propose an architecture that aims to address the aforementioned issues focusing on the on-demand delivery of multimedia documents, the synchronized presentation of them, the management of user interactions and the necessary resources. Our prototype implementation also faces the problem of interoperability by defining generalized service interfaces which are protocol and network independent.

2. Service Architecture

The architecture of the proposed interactive distributed multimedia service consists of three basic parts the *clients*, the *access servers* and the *content servers*.

The functionalities and characteristics of the basic parts of the proposed interactive distributed multimedia and hypermedia service are the following:

- **Clients**. They manage the presentation of the requested documents, ensuring the synchronization among the various involved media, the adjustment of presentation options and the necessary resources. They also communicate with the content server to report synchronization inconsistencies and to forward the interactions of the users (retrieve document, navigate, etc.). Moreover, they also allocate and manage the needed resources.

- **Access servers**. they provide a diverse set of service interfaces which are managed in an interoperable and secure way. These interfaces manage the transactions requested by clients before authentication or subscription primitives are invoked. In addition to the above, access server handles diverse client requests according to various information retrieval protocols.
- **Content servers**. They maintain the needed resources, retrieve the appropriate data from the corresponding databases, manage the transportation of the hypermedia document and the media involved in the presentation scenario, and adjust the transport rates of every single media. Content servers consist of two distinct modules, namely the *multimedia server* and the *media servers***.** The hypermedia multimedia server manages the transportation of multimedia scenarios, co-ordinates the transport process of each media by adjusting the transport rates, and also ensures the fair allocation of resources to the various connected clients. The media server manages the transportation of the relevant media streams and adjust the transport rates in order to smooth out network delays and synchronization anomalies.

The users interact with the service using a well known set of functions that are provided by the service in a transparent and friendly way. Initially, a client is connected with one particular access server. The access server upon request authenticates the client, and if the user is an authorized one, the connection can be done. From now on, the user may submit queries concerning various subjects of interest using a generalized and interoperable interface. The internal details of the used searching procedures or the specific details of the protocol used are hidden from the user. The documents-items which match the submitted query are transmitted to the client and presented to the user. The user may select one document for viewing, and in that case the appropriate content provider in which that item is resided on is conducted. Content servers, upon request, retrieve the requested hypermedia document from the hypermedia/multimedia database, transfer it to the client, and facilitate the proper transportation of the inline media objects, taking into consideration the spatio-temporal relationships among them. Media objects are retrieved from the corresponding databases and transported through different parallel connections to the client. Adjustment of the transport rates of each media is taking place continuously considering periodic feedback reports generated by the client. Based on these reports, the transportation of every media is adjusted to smooth out network delays and synchronization anomalies. The inline media objects are buffered temporarily in the client, and then they are presented according to the presentation scenario.

Access servers, clients and content servers communicate via a broadband network. Multimedia documents are stored in specialized hypermedia databases attached to the multimedia/hypermedia servers while the inline media objects reside on specific databases attached to the media servers which may be seen as networked distributed databases. Every hypermedia document is described using a representation language. This description specifies the involved inline media objects, the location of them in the corresponding media database, the way that they will be presented to the user and the synchronization relationships among them. This description is actually the presentation scenario. Documents can be linked via hyperlinks, forming a web of documents. Linked documents may be stored in the same or in different multimedia servers.

3. Client Design Issues

The main functions that the client provides to the user are the *connection* with the service, the *retrieval* of a specific item of interest and the *browsing* of the selected document. The characteristics of these components relate to the media stream management, the local buffering management, the synchronized presentation of media objects according to the presentation scenario, the resource management and the accommodation of user interactions. In Figure 1, the functional components of the client and their interactions are depicted.

Initially, users request to connect with a particular access server specifying the access code. The access code is a unique code that characterizes the specific client. Access server, using this code, authenticates the user, searching in a database of authorized users. For every client, an entry in the database of authorized users is maintained. In this database, every entry contains personal information about the user (e.g. name, address, telephone number, credit card id, etc.) and information concerning the interests of the particular client (e.g. hobbies, professional interest etc.). Moreover, an entry contains accounting information, detailed reference of the time logged in the system and documents that are retrieved in the past, as well as pricing information concerning these retrieved documents. A list of hyperlinks to the previous retrieved documents and programs is stored locally in the access server. Security and privacy is ensured using an access control list that adequately specifies which user have privileges to access the specific item of information. For every topic or information a set of permissible users, that have access to the resources, is explicitly specified

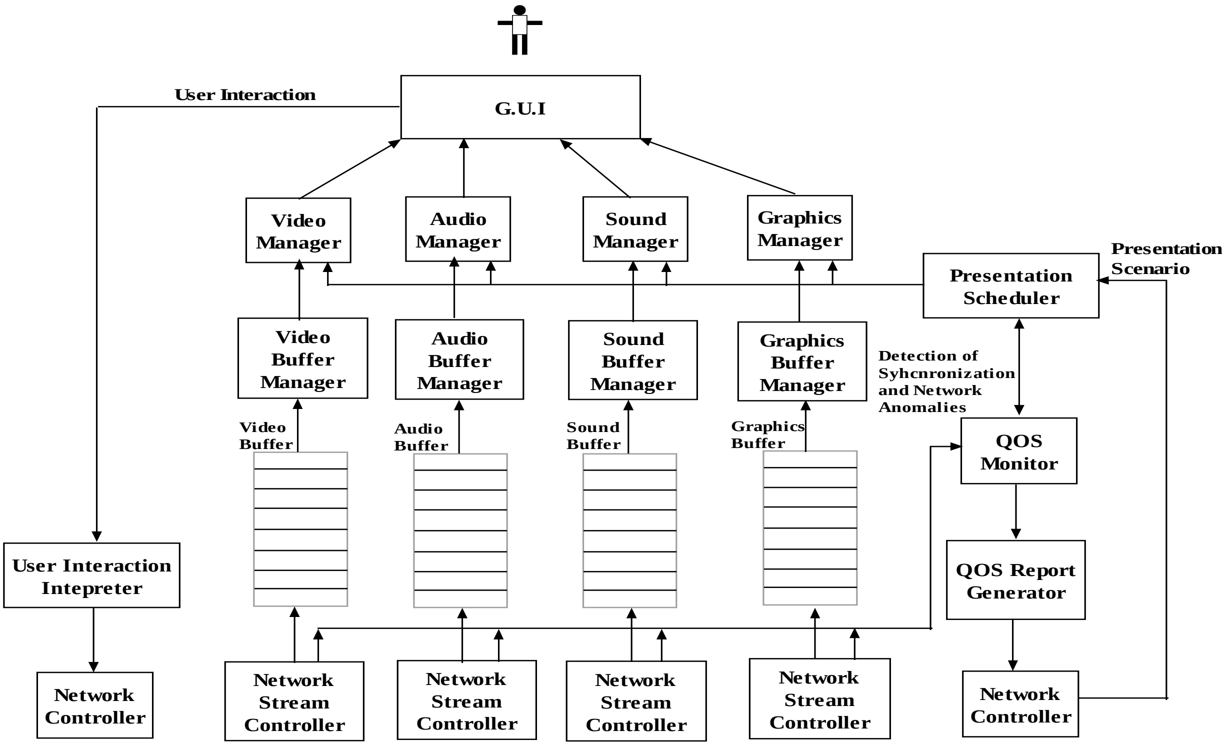

Figure 1: Client Design Schema

.After the authentication of the user and the connection establishment, users can access the *index* database in order to search for a topic of interest or request for a specific document from a certified content provider. Certified content providers should guarantee the quality of provided services, the integrity and the originality of the information. The index database contains information about various topics and pieces of information. Entries in the index database specify in which content server this particular piece of information resides on, how much it costs to access this information, whether or not that particular user is allowed to access it, and a small description about the whole hyperdocument. Only the description of the hyperdocument and the fee to access it is presented to the user, while the other fields in the entry are used internally by the access server to facilitate the request.

When a user requests a specific hyperdocument, the corresponding content server, where the requested information is resided on, is contacted. Content server, upon request, retrieves from the hypermedia database the particular presentation scenario, forwards it to the client and updates the *account database* charging the client for the requested information. Every entry in the account database contains information about the requested document, the access server that requested it, the time and date this particular request is issued and the corresponding charge.

The client, based on the presentation scenario estimates the appropriate resources concerning the network and the media buffer management. Particularly, the presentation scenario is traced and the referred media objects are determined. For every involved media, the necessary *Network Stream Controllers(NSCs)* are initialized while the corresponding media buffers are reserved. NSCs facilitate the incoming packets of every media stream, while media buffers temporarily store the received packets before they are presented to the user. The size of each media buffer is suitably calculated using several parameters, such as the network delay, the packets length, the media encoding standard and the required quality of presentation. The length of media buffers corresponds to a playback time for each media object and we define this time as the *media time window*. The media time window is used to smooth out delays inserted by the network, the operating system and other factors. In this way, the experienced delays affect (decrease) the specific media time window before affecting the quality of the presentation and the synchronization.

For every time-dependent media object with specific presentation constraints, a parallel stream session is established between the client and the appropriate media server. Data packets from a particular media object are transferred to the client through the parallel session and are received by the corresponding NSC. Received packets are directly moved to the media buffer, and “wait” until the proper time instant of their presentation is reached. NSCs and *Media Buffer Managers (MBMs)* constantly monitor the media buffer occupation and in conjunction with the *Presentation Scheduler* (which monitors the synchronization anomalies and presentation inconsistencies) extracts feedback information which is combined with information about connection parameters (delay, jitter, packet loss). Connection parameters are suitably computed by information which is embedded in

the header of the delivered packets. QoS reports, that are periodically transferred to the content server, contain all these statistically measured parameters. These reports actually reflect the quality of delivered and presented information. By examining these reports, the hypermedia/multimedia server adjusts the transport rates and the quality of each media object that is going to be delivered.
The presentation of the requested multimedia document is managed by the Presentation Scheduler in conjunction with the *Media Managers* and the MBMs. After the pre-processing of the received presentation scenario, the resource allocation, and the initialization of specific functional modules, the *playout deadlines* for every referred media object are determined. The playout deadline of every media object specifies the presentation start time of this particular media object. Furthermore, the duration of the presentation, the temporal and spatial relations among media objects are determined. Using this information, the Presentation Scheduler can arrange the presentation of each media stream according to its playout deadline. Before the presentation of a multimedia document is started, there is a “set-up” time to feed each media buffer with an initial amount of data. This delay is used to overcome network inconsistencies and manage user interactions concerning the presentation of the document. For each media object in the presentation scenario, a parallel presentation process, based on the playout deadline, is created. This process is activated whenever the presentation start time has reached. In the sequel, media objects are retrieved from the media buffers and fed at nominal rates to the corresponding Media Manager, where they are played/presented by the proper device.
The user interactions interrupt the presentation of the requested document, and initialize the appropriate actions. User interactions are either *presentation specific* or *document specific*. Presentation specific actions activate either the re-start of the presentation or the backward presentation. In both cases, the Presentation Scheduler carries on the presentation of the document from the requested time instant, using data from the media buffers and hence, no presentation gap is experienced. The size of the media buffers corresponds to a presentation time interval equal to the propagation delay time needed for the delivery of the next packets of every media object. When the Presentation Scheduler “reaches” the end of the media buffer, the next media objects that should be presented have already arrived at the client. Document specific interactions may be: return to the previous presented document, follow a hyperlink etc., and are managed in the same way as in the case of the first requested document. These interactions can be triggered any time during the presentation, causing the presentation of the current document to stop.

4. Access Server Design Issues

The main functions that the access server provides are the *authorization* of the user, the *subscription* to the service, *search* capabilities and the *transmission* of the results of the requested queries. The access server provides a generalized interface to the clients according to which various information retrieval protocols are embedded, while certain details are hidden (e.g. HTTP, SQL, etc.).
Initially, a client tries to connect with a service in order to have access to the information which is provided by in various content providers. The access server, upon request, invokes a authentication primitive. According to this

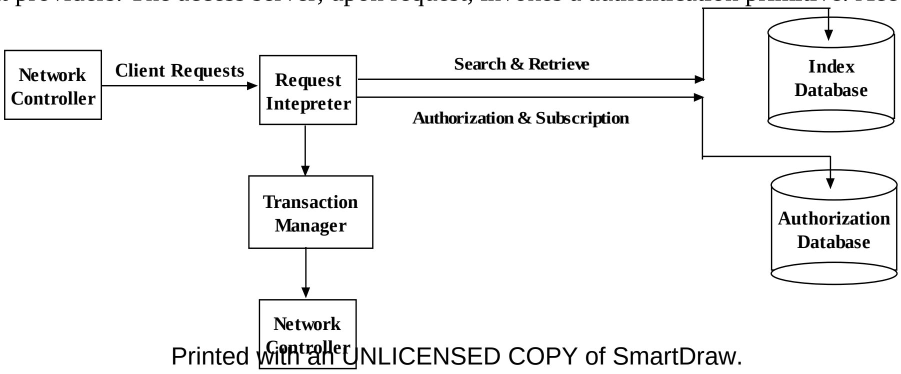

Figure 2: Access Server Design Schema

primitive, a coherent centralized database is searched in order to ensure that the user can have access to the service. In case that the user is not a authorized one a subscription primitive is invoked. The user fills in standard form and submits it to the access server. A new entry in the database of the authorized users is created, while a unique access code is sent to the user. This new entry is broadcasted to every co-operative access server in the service in order to obtain coherence to the contents of the authorized database.
After the connection establishment the user may submit queries to the access server in order to obtain information on specific thematical units. The access server searches locally the index database in order to find all the relevant items that satisfy the requested query, while also the same query is broadcasted to all the co-operative access server for the same reason. This mean that the index database is a distributed database at which all access server may retrieve the necessary items. The results of the query are forwarded to the specific access server which

triggers the procedure and then forwarded to the appropriate client. Using this schema, information from different content providers may be added locally in the index database of one access server without causing any additional overhead. Every item which matches the requested query actually contains *meta-data* information concerning the actual document which describes. In addition to that, meta-data information determines the originator of the multimedia document and the content provider at which that document is resided on. Using this information the client can directly request the document from the corresponding content server. In figure 2, the functional modules of the access server are illustrated. *Network Controllers* facilitate the network specific actions, while the *Request Interpreter* manages user interactions. The *Transaction Manager* manages the appropriate transactions with the other co-operative access servers according to the previous described procedure.

5. Content Server Design Issues

The main functions of the content server is the accommodation of document requests, the charge of requested information, and the resource allocation and management of the needed resources. Requested documents are delivered on-demand while the whole transport process is being adjusted continuously to keep the service quality in efficient levels.

When a request is issued, the multimedia server estimates the resources needed for the transportation of the documents and computes the flow scenario. The flow scenario specifies the inline media objects, when the transportation should start, and which the transport rate should be. The flow scenario is computed using the presentation scenario, considering the spatio-temporal relationships among various media, the current network conditions, and the available resources in the access server. Every media server manages the appropriate resources using the flow scenario.

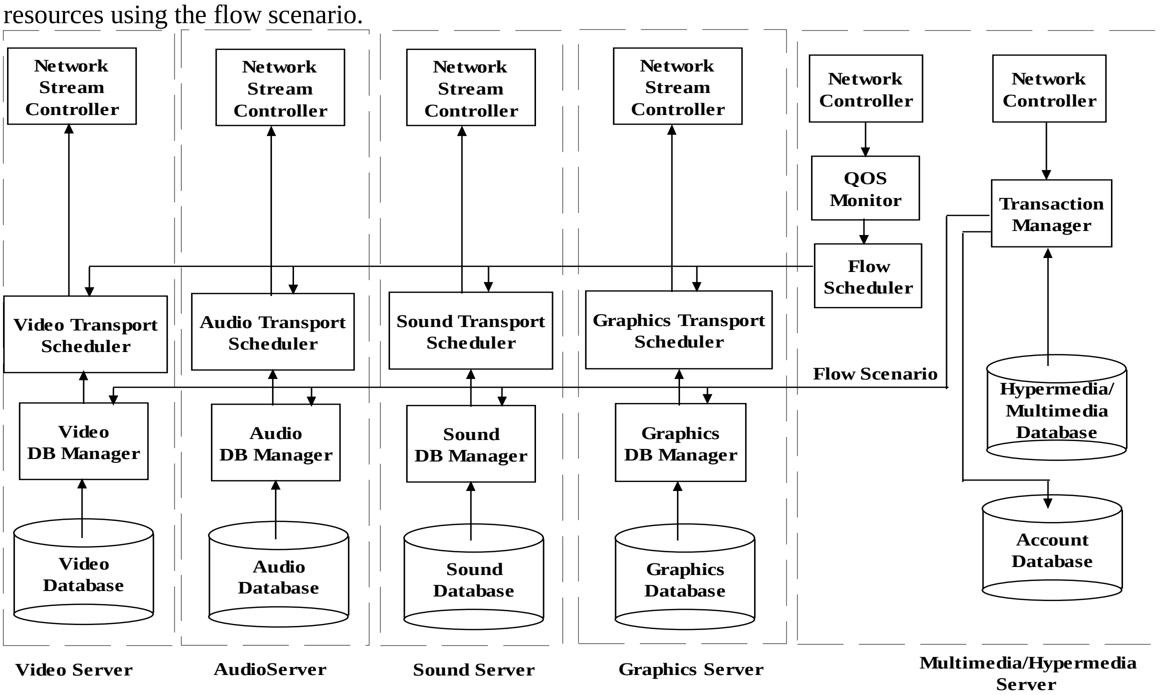

Figure 3: Content Server Design Scheme

Every media server, based on the flow scenario, retrieves the corresponding media objects from the media database, splits the data into packets and according to a well defined transmission rate transfers them to the appropriate access server through a dedicated parallel connection. Adjustment of the transmission rate of each media is taking place in order to meet the presentation constraints and network conditions. The adjustment is done using periodic feedback reports generated by the client (QoS reports).

User interactions, concerning presentation operations, interrupt the current transportation process and cause a re-computation of the flow scenario. The exact time instant, at which the presentation of the document will be re-started, is determined. Based on buffer sizes of the client and access server respectively, the exact media objects and their real-time constraints are also determined. In the sequel, the transportation process is re-started according to the new flow scenario. Using this approach, the gaps experienced in the presentation are minimized. Figure 3 illustrates the functional modules of the Content Server.

6. Implementation Issues

The proposed design approach was implemented and a prototype version that satisfies the above mentioned characteristics and capabilities is already developed. The service is tested over a 100 Mbps FDDI ring using

TCP/IP protocol suite. Access and content servers are running on various Unix Workstations while the clients running on PCs. Clients can be either directly interconnected to the FDDI ring or interconnected in a LAN which is interconnected with the FDDI.

Multimedia documents are explicitly described using a hypermedia markup language which supports the necessary capabilities. The markup language offers a set of primitives for the presentation and the synchronization of the inline media objects and the interconnection among documents. A run time environment is specially implemented in order to interpret and facilitate these primitives.

For the transmission of data, we make use of the *Real-Time Transport Protocol* (RTP). The RTP is an intermediate protocol that provides an application with end-to-end functions for transmission of real-time data, such as audio and video. RTP is augmented with the control protocol Real Time Control Protocol (RTCP), which enables several actions, such as monitoring of the parameters of an ongoing session. RTP does not quarantee QoS in data transmission, and relies on network protocols to make use of their transmission primitives. We use this packet’s header to derive statistical measurements concerning the network conditions (jitter, packet loss, etc.).

7. Conclusions

We presented an architecture for interactive distributed multimedia information services. Various issues concerning the on-demand delivery of multimedia documents, resource allocation and management are presented. Moreover, issues concerning the synchronized presentation of multimedia documents and the management of user interactions are discussed. The performance of our prototype implementation is quite good due to the dynamic allocation and management of needed resources. In addition to that, the adjustment of the on-demand delivery process according to the network conditions is also constitute to the achieved performance. Another key issue is the level of scalability and flexibility which is rather high due to the distributed nature of the service.

We intend to expand the proposed architecture to support on-line billing mechanisms and better resource management policies. We also intend to incorporate in the prototype version QoS quarantee primitives. Currently, we are on the phase of analysis and specification of a protocol for the on-demand delivery of multimedia documents over ATM-based networks. In addition to that, we intend to port our prototype implementation in Corba-based platforms, while also investigate the potential to incorporate the requirements and specifications of DAVIC.

8. References

[1] D. P. Anderson G. Homsy, "A Continous Media I/O Server and its Synchronization Mechanisms". Computer, Vol. 24, No. 10, October 1991, pp. 51-57.

[2] Blakowski, Steiwetz, "A Media Synchronization Survey: Reference Model, Specification, and Case Studies". IEEE Journal on Selected Areas in Communications Vol. 14, No 1, pp. 5-35.

[3] C.Bouras, V. Kapoulas, D. Miras, V. Ouzounis, P. Spirakis, A. Tatakis, "On-Demand Hypermedia Multimedia Service over Broadband Networks". Fifth IEEE International Symposium on High Performance Distributed Computing, Focus Workshop on Multimedia Collaborative Environments, 6-9 August 1996, Syracuse, New York, USA.

[4] T.D.C. Little, D. Venkatesh, "Client-Server Metadata Management for the delivery of Movies in a Video-On-Demand System". In Proc. 1st International Workshop on Services in Distributed and Networked Environments, Prague, Czech Republic, June 1994, pp. 11-18.

[5] R. Ramorao, V. Ramamoorthy, "Architecture Design of On-Demand Video Delivery Systems:The Spatio-Temporal Storage Allocation Problem". In Proc. ICC 1991, pp. 17.6.1-17.6.5

[6] P. V. Rangan, H. M. Vin, S. Ramarathan, "Designing an On-Demand Multimedia Service". IEEE Communication Magazine, Vol. 30, No 7, July 1992, pp. 56-64.

[7] R. Steinmetz R., K. Nahrstedt, "Fundamentals in Multimedia Computing and Communications". Englewood Cliffs, NJ,Prentice-Hall, 1995.