# ON DEMAND DELIVERY OF MULTIMEDIA DOCUMENTS USING DISTRIBUTED OBJECTS

C. BOURAS[1,2] V. OUZOUNIS[3] P. SPIRAKIS [1,2]

[1] *Computer Engineering and Informatics Department, University of Patras,*
[2] *Computer Technology Institute, Patras, Greece*
[3] *GMD Fokus, Berlin, Germany*
e-mail: bouras@cti.gr

## Abstract

In this paper, we propose an architecture for the on-demand delivery of multimedia documents over broadband networks using distributed objects. The basic components of our model are the *network agents* and the *presentation agents*. Our model preserves flexibility, transparency and modularity and value added opportunity due to the usage of distributed objects.

**Keywords:** Computer Networks, Distributed data, Hypermedia/Multimedia

## 1 INTRODUCTION

During the recent years we are experiencing a significant development of the multimedia technologies and a rapid progress in the network infrastructure. These developments make feasible the delivery of multimedia information among sites over the network and contribute to the emergence of new applications that involves multiple media objects. Such applications may be used for distance education, remote experting, remote access to digital libraries, multimedia news services, electronic magazines, entertainment and many other. These applications are going to become the prime consumers of network resources [1, 2]. As the successful introduction of multimedia applications will depend on the user acceptance of "quality" and on the cost of services, an evolutionary approach to multimedia services is essential.

Advanced multimedia systems are characterized by the integrated computer-controlled generation, storage, communication, manipulation and presentation of independent time-dependent and time-independent media

A distributed multimedia information system can be considered as a set of functions or entities 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 an integrated and unified way. 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, 3]. 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 [5].

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, modular and interoperable interfaces to diverse information suppliers, various encoding schemes and multimedia representation methods, diverse protocols and network platforms and various device characteristics [4].

In this paper, we propose an architecture for the on demand delivery of multimedia documents over broadband networks using distributed object technology. Our model aims to address the diversity of various involved parties in these services, providing value added opportunity, in the sense that new services can be easily incorporated, and modularity, in the sense that basic operations are defined in modular manner with each service element responsible for a specific role.

## 2 THE PROPOSED ARCHITECTURE

The proposed architecture consists of three layers, the *service layer*, the *presentation layer*, and the *network layer*. In every layer, key components with specific distinct role and functionality are identified. These components are referred to as objects with well defined interface. Therefore, the proposed model can be considered as a *set of distributed objects* with *well defined interactions*. An object interacts with another object via a message-based interface, regardless if these objects are located in the same or in different sites. Whenever a message is sent, the appropriate method is invoked, in order to facilitate the request. Our model is presented graphically in Figure 1.

The service layer provides the necessary operations from the user point of view and the information provider. All the details concerning the accomplishment of these operations are hidden. In the ideal case, general interoperable interfaces should be supported in order to satisfy various service requirements and user needs [5].

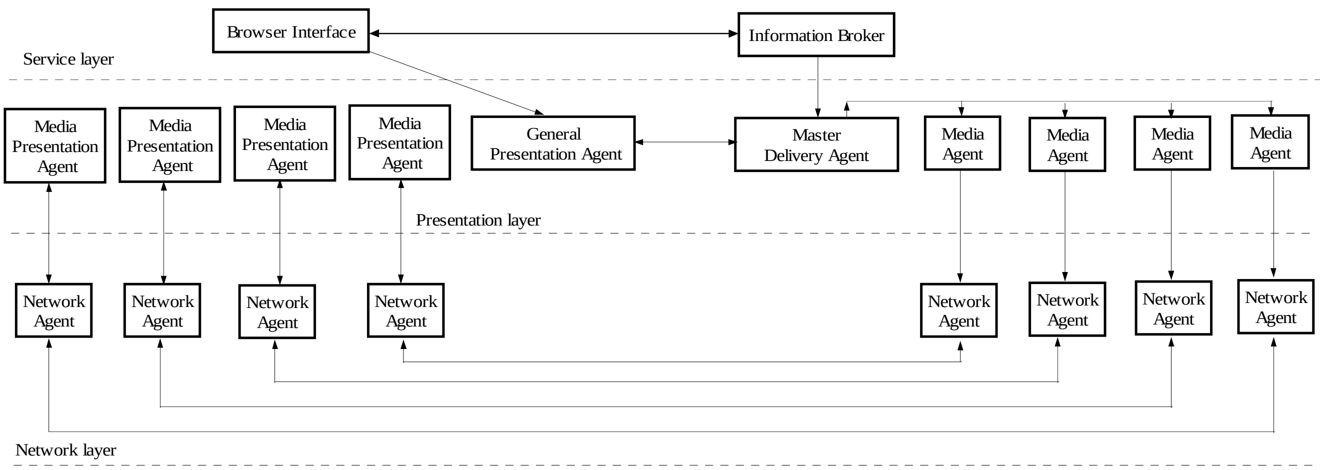


**FIGURE 1:** THE PROPOSED ARCHITECTURE

The presentation layer provides operations concerning media retrieval, and media presentation. Finally, the network layer provides the necessary operations for the on-demand delivery of media objects. Interoperable interfaces which support diverse network platforms and protocols must be included. Distributed object technology can effectively support these needs using inheritance, polymorphism and encapsulation.

At the service layer, two are the key objects, the *Browser Interface* (BI) and the *Information Broker*(IB). The Browser Interface is a well defined interface that provides the user with the ability to connect to a service, submit a query, and select a multimedia document for browsing. During the presentation of the document, the user may interact with the document, e.g. to reload the document, to pause presentation, etc. The Information Broker authenticates the user before the connection establishment phase. If the user is an authorized one, the Information Broker replies to BI's queries and transmits the matching items. In addition to that, whenever the user selects a document for browsing, IB triggers the on-demand delivery process.

At the presentation layer, whenever a document is selected the *Master Delivery Agent* (MDA) locates the selected document and, by computing the playout scenario, determines the involved media objects, and their spatio-temporal relations. The playout scenario actually specifies the exact media objects and when these media objects should be played. For every involved media object in the playout scenario, the corresponding *Media Agent* (MA) is conducted and the appropriate information (e.g., media object id, time constraints, etc.) which are used to locate the appropriate objects are passed as parameters. The Media Agents are, in nature, distributed objects and, thus, can be situated in different sites. This means that media objects as well as Media Agents can be located in different sites. The distributed media objects are transmitted to the receiver in a transparent way, i.e., without the receiving site noticing the distributed delivery process. The Media Agents locate the needed objects and forward them to the *Network Agents* (NA). Network Agents at the transmitting site, are responsible for the connection establishment and the transmission of media objects. Information regarding media encoding schemes and multimedia representation formats are hidden by the Network Agents, which face media objects as raw data with specific time delivery constraints. Therefore, a Network Agent can be considered as a general purpose delivery mechanism.

Particularly, the NA's at the transmitting site, after the connection establishment process, negotiate for the needed network resources and deliver the appropriate media objects on time according to the strict timing constraints. Network and protocol specific details are also hidden, and as a result transparency is achieved.

At the receiving site, Network Agents receive the transmitted media objects and forward them to the *Media Presentation Agents* (MPA's). Media Presentation Agents playout the media objects according to their time constraints and spatio-temporal relations. The MPAs encapsulate various encoding schemes and diverse device characteristics in order to increase flexibility and generality..

In addition to that, our model can effectively support user interactions during the presentation. In particular, the user may desire to pause presentation, to re-start presentation, to stop presentation, etc. In that case, the BI notifies the General Presentation Agent about user selection. For example, when the desires to stop presentation, he/she triggers the appropriate selection. The BI according to the user selection, informs the GPA which undertakes the responsibility to accomplish the task. The GPA informs the involved Media Presentation Agent to stop presentation, while also communicate with MDA. In the sequel, MDA informs Media Agents to stop delivering the media objects.

## 3 TRANSACTION-PROCESSING MODEL

In this section we will describe, in detail, how the flow of messages between objects is achieved. In particular, there are ten major phases that they will be presented in turn, giving the processing that occurs and the corresponding messages that are sent between objects. The notation BI→IB.start_session(BI) means that the Browser Interface sends the message *start_session* to the Information Broker. The argument BI in the message can be seen as a parameter. In table 1, the abbreviations, which are used for all the referred objects, are given. The names of the messages which are sent between objects are suitably chosen in order to denote the operations which are performed.

| Symbol | Object |
|---|---|
| BI | Browser Interface |
| IB | Information Broker |
| MDA | Master Delivery Agent |
| GPA | General Presentation Agent |
| MD | Media Agent |
| MPA | Media Presentation Agent |
| NA_D | Network Agent at the transmitting site |
| NA_R | Network Agent at the receiving site |

**TABLE 1:** NOTATION TABLE

### A. Connect to the service

Before the user submits queries and retrieve multimedia documents for browsing, he/she must connect with an information provider. In that case, a *start_session* message is sent to the Information Broker. The service authenticates the user and establishes a session with a unique *session_id*. In case that the user is not an authorized, the service does not allow the BI to connect with the service and the reason for that is given.

### B. Submit a query

When the BI is connected with the service, queries can be freely submitted. The BI sends a submit_query message with the specified search conditions. Details concerning the retrieval process are not included.

### C. Results of query

The results of the query are sent to BI using the message *results_of_query*. In that message, the parameter *results* contains all the matching items. For every item, the corresponding *item_id* is included. The *item_id* has semantic meaning only to IB and represents the specific document.

### D. Select a multimedia document

When an item is selected for browsing, a retrieve item message is sent. In that message, except from the *session_id* and the *item_id* parameterer, a pointer to MPA is specified.

### E. Trigger on-demand delivery process

When the documents is selected, the IB informs asynchrounsly the MDA about the event.. From now on, the MDA undertakes the responsibility to manage the on-demand delivery of the selected document. The IB can serve new requests from other Bis, while also can not have any control on the delivery process. The MDA sends to the GPA the playout scenario, in order to determine the involved media object and dynamically specify the needed MAs which will receive by the network the transmitted data. NAs have common functionality and well defined interface. In addition, they face media objects as raw data with specific time constraints. Therefore, NAs are in general media, network and protocol independent objects. The parameter *playout_scenario* actually specifies the involved media objects and their spato-temporal relationships.

### F. Master Delivery Agent contacts involved Media Agents in order to trigger media object retrieval and transmission procedures

When the MDA gets the list with the appropriate agents, conducts the involved Media Agents in order to trigger the retrieval and the transmission of them. All the responsibility for the delivery of media objects is passed to the corresponding Media Agents. These agents are distributed in nature and serve as generalized retrieval and delivery mechanisms. The parameter *media_object_id* specifies the media object, while the *time_constraints* determine the strict presentation constraints.

### G. Every Media Agent contacts a Network Agent to facilitate network transmission of the corresponding media objects

When the media object is detected, the transmission process is triggered. For that reason, a Network Agent is informed in order to start the transmission.

### H. Every Network Agent establishes connection with another network agent at the receiving site and negotiate the needed resources

Before the transmission of media objects, a connection between the two network agents must be built. During the connection establishment process negotiation between them can take part in order to reserving the needed resources.

### I. The Network Agent at the transmitting site starts the delivery of media objects according to the specified time constraints

After the connection establishment process, the media objects can be sent. The transmission of them is done continuously based on the time constraints and network resources which are previously allocated. Buffer mechanisms are not involved since these operations are encapsulated inside the network agent functionality.

**J. The Network Agent at the receiving site forwards the collected data to the appropriate Media Presentation Agents which playout the media objects**

When the batches of media objects, are retrieved by the MAD, they are directly feed to the MDA. The MDA's are generalized presentation agents which can present various media objects using the same operations. This can be achieved, using polymorphism. Consequently, media encoding details, and device specific characteristics are hidden.

In addition to the above functions, our model can effectively manage network anomalies, synchronization anomalies and user interactions. In the first case, whenever the network agent at the receiving site detects a time constraint violation, informs the network agent at the transmitting site about the event. In that case, the two involved parties may re-negotiate for new network resources. In the second case, whenever a synchronization anomaly occurs, the MDA, which detects the event, informs the Network Agent about it. Then, the same procedure as in the first case is followed.

## 4 DISCUSSION

A prototype version of the proposed model has been already performed. This implementation is based on a public free version of CORBA[8]. Particularly, we used ILU distributed system [9] as an object oriented platform. The proposed model was implemented in C/C++. Actually, the proposed model is a natural evolution of a previous on-demand multimedia distributed system which was implemented for distance education purpose. We implement a generalized interface to this service, and we intend to extent this interface to the WWW for experimental reasons. The implementation process focuses on two directions. The first direction is to develop a CORBA based on-demand delivery model using customize approaches, while the second one is to inherit our model in a DAVIC [6,7] compliant system. Particularly, we currently investigate the potential interfaces that should be specified and how can we empower our model inheriting well-defined DAVIC interfaces..

## 5 CONCLUSIONS

In this paper, we proposed an architecture for the on-demand delivery of multimedia documents. Our model can be seen as a set of distributed objects with well defined general purpose interfaces. These interfaces are generalized in nature and can support the increased diversity in the area of distributed services. The interactions among objects are considered as messages which trigger the appropriate actions to be performed. Our model, due to the usage of distributed object technology, preserves flexibility, transparency, modularity, value added opportunity, and a tolerable level of interoperability.

## References


[1] Cosmos Nicolaou, "An Architecture for Real-Time Multimedia Communication Systems", *IEEE Journal on Selected Areas in Communications*, Vol.. 8, No. 3, Apr. 1990

[2] P. Venkat Rangan, Harrick M. Vin, and Srinivas Ramanathan, "Designing An On-Demand Multimedia Service", *IEEE Communications Magazine*, Jul. 1992, pp. 56-64

[3] Yahya Y. Al-Salqan, "MEDIAWARE: On Distributed Multimedia Synchrronization", CERC-TR-RN-95-007

[4] R. Kahn and R. Wilenksky, "A Framework for Distributed Digital Object Services", Tech. Report cnri.dblib/tn95-01, Reston, Va, 1995; http://www.cnri.reston.va.us/cstr/arch/kw.html

[5] A. Paepcke, S. Cousins, et al. "Using Distributed Objects for Digital Library Interoperability", *Computer*, May 1996, p. 61-68

[6] Digital Audio Visual Council (DAVIC), "DAVIC 1.0 Specification Part 03", http://www.davic.org/

[7] Digital Audio Visual Council (DAVIC), "DAVIC 1.0 Specification Part 04", http://www.davic.org/

[8] Object Management Group (OMG). The common object request broker: Architecture and Specification, revision 2.0. 1995, ftp://ftp.omg.org/

[9] A. Courtney, W. Hanssen, et al., Inter-language unification, release 1.5. technical report ISTL-CSA-94-01-01, Xerox PRC, May 1994. Available at ftp://ftp.parc.xerox.com/pub/ilu/ilu.html