Distributed Multimedia Information Presenting Environment

Shoou-Gwo Jiang Gin-Kou Ma

Advanced Technology Center
Computer and Communication Laboratories
Industrial Technology Research Institute
Chutung, Hsinchu, Taiwan, R.O.C. 310

Abstract

Many multimedia systems are focusing on the techniques of teleconferencing applications and discussing the general reference architectures. Generally speaking, a distributed multimedia application is constructed of four layers: the interactive layer, the data composition layer, the network transport layer and the database management system (DBMS) layer. Although these layers are the well-known reference architectures, the connections (multimedia modeling, module operations, and module communications) between them still need to be solved. To allow users easily, flexibly, and dynamically to create, query, and present multimedia documents, we propose five modules as an integration solution; they are, interactive user interface module, multimedia object manipulation module, object dynamically mapping module, network transport control module, and physical object file manipulation module. These modules are the essences of the behaviorally object-oriented distributed multimedia environment. Based on these modules, the multimedia object modeling, the query processing and the presentation schema will be elaborated.

1 Overview

Techniques for teleconferencing applications [1] and a few general reference architectures about multimedia systems have been developing [2]. On the whole, a multimedia application is constructed of three layers [3]: the interactive layer, the data composition layer, and the DBMS layer. Meanwhile, for a distributed multimedia application, the network transport layer should be added. Although these layers are the well-known reference architectures, there are still lots of problems must be solved, such as, how to satisfy users' views by the sophisticated user interface, how to manipulate multimedia objects, how to compose various multimedia objects as a multimedia document and present it, how to swap multimedia object consistently between temporary state (run-time memory) and persistent state (object database), how to achieve location transparency (files and users) and control data stream on a network, and how to map a multimedia object from an object database to a physical storage database. To solve these problems, we first define a multimedia object as one of formatted text, graphics, image, digitized audio or video, and multimedia document. The multimedia document object includes a presentation schema with layout data and scheduler of multimedia objects. The multimedia object model is the core part of the distributed multimedia environment (DME). It affects the performance and efficiency of the designs of the interactive user interface, the presentation schema, the query methodology, the network transport control, and the object database management.

For easily, flexibly, and dynamically creating, querying, and presenting multimedia documents, five modules are then proposed as an integration solution for the above addressed problems. They are the interactive user interface module, the multimedia object manipulation module, the object dynamically mapping module, the network transport control module, and the physical object file manipulation module. These modules are the essences of the behaviorally object-oriented distributed multimedia environment. After having the modules been defined, the designs of the multimedia object model, query processing and presentation schema will be detailed as follows.

2 Modules For The DME

2.1 Interactive User Interface Module

The Windows system (such as X-Windows, MS Windows et al.) that provides multi-windows and multi-tasking capabilities is the state of arts for designing interactive user interface. For DME, it should be carefully considered about how to guide users manipulating naturally and how to attract users enjoying themselves thoroughly. Therefore, it is no doubt for us to adopt the human factor methodology to design the appropriate user interface with the Windows system modules.

2.2 Multimedia Object Manipulation Module

The multimedia object manipulation module has three parts: the multimedia object model, the multilevel query methodology, and the presentation schema creation. Now we briefly elaborate on each of them.

Multimedia object model The object-oriented approach is adopted for modeling the multimedia ob-

0-8186-2755-7/92 $03.00 © 1992 IEEE

jects. In the DME, a multimedia object is the basic distributed object unit, called distributed multimedia object. In accordance with the various properties of multimedia objects, we apply the ADT (abstract data type) concept to define attributes and methods for each multimedia object. The *Network_Transport* class and the *Features* class, called supertypes, are created for the general property of the network transporting and the querying. For the specialization purpose, six multimedia object classes, called subtypes, are derived from the *Network_Transport* class and the *Features* class; they are the *Formatted_Text* class, the *Graphics* class, the *Image* class, the *Audio* class, the *Video* class, and the *Multimedia_Document* class. Through the multiple-inheritance mechanism, these multimedia objects own the network transport capability and the specific feature functions as shown in Figure 1.

Querying Most multimedia systems use the simple keywords/sentences matching, or the complicated SOOSQL (Semantic Object-Oriented SQL)-like [4], Lisp-like, or other language for the query processing. However, because of the simplicity the keyword-base approaches lower the information retrieval efficiency and effectiveness. On the contrary, the SOOSQL-like or Lisp-like language evokes too much implementation overhead/cost due to its complexity and is user unfriendly. To eliminate these negative effects and to conform to the DME requirements, we propose a new query methodology, called *multi-level query*, which is detailed in Section 4.

Presentation schema creation For the multimedia presentation under DME, a presentation schema is essential. A presentation schema consists of two phases : *layout phase* and *schedule phase*. The layout phase concerns the manipulations of selection, multilevel query, location, and window-size assignment, and records the resulted attributes for each distributed multimedia object, see Figure 2(a). The schedule phase concerns the arrangement and the timing of all selected distributed multimedia objects. The *time-resources scheduler* is designed for the schedule phase. As shown in Figure 2(b), the whole presentation is called one *scene*, in which #1, #2, and #3 stands for *section I*, *section II*,and *section III*, respectively. The *section* means the appropriate timing segmentation.

2.3 Object Dynamically Mapping Module

A multimedia object is processed in two states: temporary and persistent. When a multimedia object is loaded or created into a run-time memory for processing, it is in the temporary state. And when the multimedia object is stored into an object database, it is in the persistent state. The object dynamically mapping module is for the swapping between these two states, that is, how to map a temporary multimedia object to a persistent multimedia object and how to map a persistent multimedia object to a temporary multimedia object. To achieve this swapping function, we design an object structure that consists of four fields: the attribute structure, the representative description structure, the physical file name of multimedia object, and the creator name. For a multimedia object transferred from a temporary state to a persistent state, the process is to fill the related field's data and store the object structure into an object database. The reverse process is to retrieve the object structure from the object database and dynamically use the *new* function (C++ function) to create a multimedia object and then set its attribute values with the retrieved object structure.

2.4 Network Transport Control Module

The function of network transport control module includes the management of the registration database, the control of the time-resource scheduler, the allocation of buffers, the packetization/depacketization of information, and the functions of transmission.

Registration database management The goal of registration database is to keep the distributed multimedia objects consistent. Once a DME user creates a multimedia document, he will encounter the consistency problem under the distributed multimedia environment. For example, the user has composed his multimedia document by using the remote multimedia objects through the network. But, the remote multimedia objects may be deleted or migrated at any time. Therefore, we propose the registration database for solving this problem. The working site (calling node) needs the called node address and the called physical object file name, which is just a part of attributes of a distributed multimedia object. In the remote site (called node), the registration database records the calling node address, the calling document file name, and the called physical object file name; these construct the registration record. In this way, whenever either the called or the calling multimedia object is changed, the registration database manager knows that one it should notify, so the distributed multimedia object can be manipulated in a consistent state.

Time-resource scheduler control A multimedia document has an associated time-resources scheduler that contains the timing and the contents' information. Although all multimedia objects have the capability of network transmission, their timing for the presentation and their transmission time may be different. Therefore, a controller to control the scheduler is required. The controller, according to the presentation time sequence, prepares the multimedia objects. The preparation timing should consider the safe_buffering time of the multimedia objects for the transmission error, the networking delay, or the unpredictable factors.

Buffers' allocation In the file transfer application, only short memory buffer is required for a long chunk transfer because the destination is a file not a memory buffer. However, in a distributed multimedia application it is totally different. The multimedia objects may be too large to reside in the memory at a time. Thus a buffer allocater is required to deal with the memory allocation according to the time-resource scheduler and the type of multimedia objects. Details of the buffer control strategy are explained in Section 5.

Packetization/depacketization and Transmission on network These functions are so obvious that it is not necessary to go into details. But, there is one thing deserved to be mentioned. In the DME, the stream synchronization of distributed multimedia objects is more seriously emphasized than the real-time requirements of the multimedia presentation. In principle, a network user could purchase the powerful video/image/audio/graphics interface cards and the high-speed network if a real-time multimedia presentation is required.

2.5 Physical Object File Manipulation Module

We have introduced the object structure in the Section 2.3, which contains four fields: the attribute structure, the representative description structure, the file name of multimedia object, and the creator name. The object structure is a basic record of the object database. For each multimedia object, we particularly design an object database, therefore, there are image, graphics, text, audio, video, and multimedia document databases. After a network user has created a multimedia object by an authoring system, he fills the information into the object structure and the multimedia object is stored into the corresponding object database. While the user want to delete or migrate a physical object file, the modification of its object database as well as registration database ,and the notification to the related network stations are followed.

3 Multimedia Object Modeling

The *Network_Transport* class, as shown in Figure 1, defines a set of data and methods for communication protocols (TCP/IP, NetBIOS, or et al.), and a set of virtual functions for sending and receiving information. The virtual function designed for different distributed multimedia objects has the same function name but deals with different procedures/fun ctions. The *Features* class defines a set of virtual functions for the layout, the adjustment, and the compression/decompression schemes, and a set of data and methods for representative description structures. Through the multiple-inheritance mechanism, all distributed multimedia objects also own them. Besides, the *Video* class defines a set of data and methods for the total time and frames, the *Audio* class defines a set of data and methods for the total time, and the *Multimedia_Document* class defines a presentation scheme structure.

4 Multi-level Querying

From the human natural point of view, we propose the multi-level query methodology for the user to query and retrieve the distributed multimedia objects friendly. The multi-level querying is shown in Figure 3. First, a user selects one type of multimedia objects and then input a list of keywords about the properties of the expected multimedia object for querying. Afterward, the query processor will search the expected multimedia object from the local object databases first. If the query processor cannot find the matched multimedia objects, it will broadcast the query request to all the remote nodes. After collecting these acknowledges from the remote nodes, the query processor refines the responses and let the user to make a decision to confirm. If the user decides to confirm, the query processor will perform the network transporting, the remote registration database updating, and a new multimedia object(s) creating, and then end the query processing. Otherwise, the user can select the preview function that the query processor will process the network transporting and show the matching objects on the screen to help the user to identify them (after that, the selected remote nodes will be notified to update their registration database), or can select the keyword-list modification function to refine the keyword-list to query again.

5 Buffer Control

Different from the buffer control for the conventional file transfer, the buffer allocation within the DME has two strategies as followes:

- The allocated buffers cannot destroy while the duration of presentation, such as image, formatted text, and graphics multimedia objects.
- The allocated buffers can be destroyed after an instantaneous presentation is finished, for example, the video or the audio multimedia object presentations.

Based on the above considerations, the maximum buffer size (MBS) of a multimedia document section show is the key index for allocating buffer. For example, from Figure 2(b), the MBS of *section II* is equal to the sum of the following: the `sizeof(one time unit of video#2)`, the `sizeof(one time unit` *of audio#2)*, the `sizeof(image#2 )`, and the `sizeof(text#2)`. If the safe_buffering time for this section is τ_{SB}, and the start time of this section is τ_S . Then to be sure that the objects reside in the buffers for smooth presentation, we must preallocate the MBS at $\tau_S - \tau_{SB}$. Under this principle, a multimedia document section show will not be pended and the memory buffer can be used efficiently.

6 Summary

In this paper, we propose five modules as the backbone for building distributed multimedia information presenting environment; they are, interactive user interface module, multimedia object manipulation module, object dynamically mapping module, network transport control module, and physical object database module. Besides, the multimedia object modeling, the multi-level querying, the presentation schema, and the buffer control are also detailed. Linking them together, we provide an integrated system that solves the connection problems between the four well-known distributed multimedia application constructive layers: the interactive layer, the data composition layer, the network transport layer and the database management system (DBMS) layer. In such a distributed multimedia environment, the user can easily, flexibly, and dynamically create, query, and present multimedia documents. However, issues such as buffer control for the distributed presentation playback fucntion, real-time heterogeneous environment, and distributed load balancing must be further considered in the future for enhancing the DME applications.

References

[1] G. Coulson, N. Williams and G.S. Blair, "Group presentation of multimedia applications in IXION", Computer Communications, VOL. 14,NO. 4, MAY 1991

[2] Wolfgang Klas, Erich J. Neuhold et al., "Using an object-oriented approach to model multimedia data", Computer Communications, VOL. 13,NO. 4, MAY 1990

[3] P.B. Berra, C.Y.R. Chen et al., "Architecture for distributed multimedia database systems", Computer Communications, VOL. 13,NO. 4, MAY 1990

[4] C. Meghini, F. Rabitti and C. Thanos, "Conceptual Modeling of Multimedia Documents", IEEE COMPUTER, VOL. 24,NO. 10, OCTOBER 1991

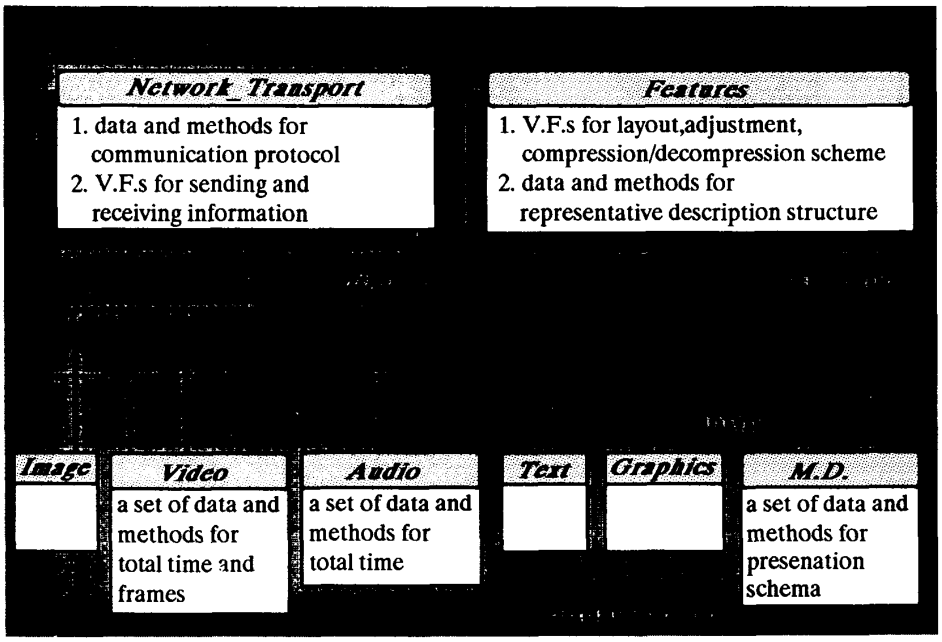

Figure1: Distributed Multimedia Object Model

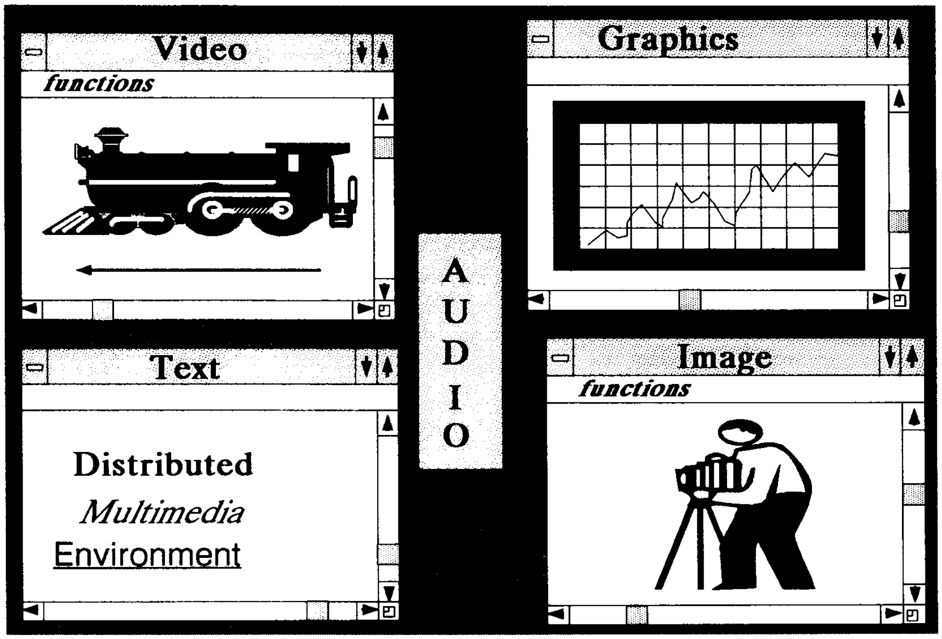

Figure2(a): An appearances of Layout phase

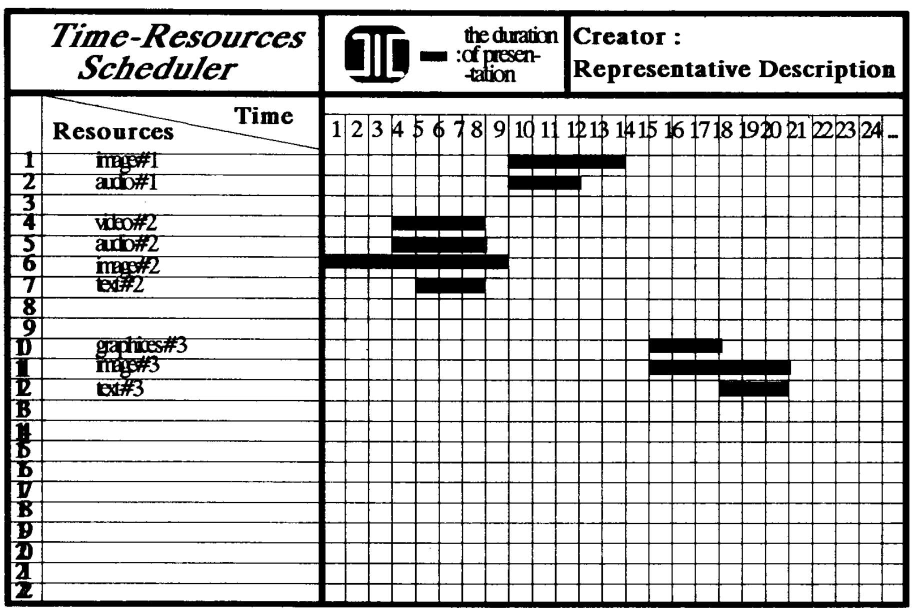

Figure2(b): Time-Resources Scheduler

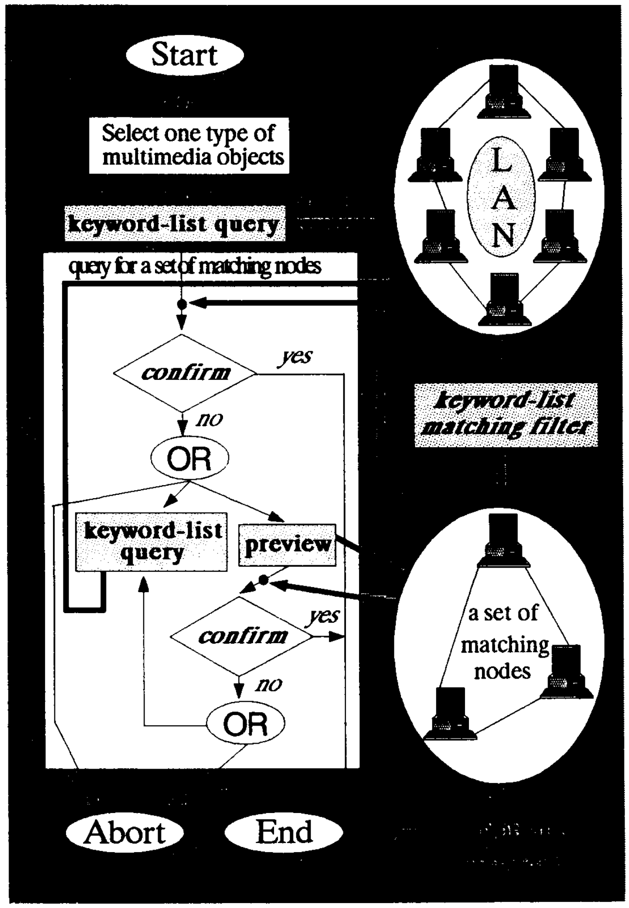

Figure3: Multi-level Query Methodology