多媒體資料庫管理系統之架構與物件整合
Architecture and Object Composition in an Multimedia Database Management System

施國琛,王慶生,王怡秋,張百福,邱川峰,戴伶娟
Timothy K. Shih, Ching-Sheng Wang, Yi-Chiou Wang, Julian B. Chang, Chuan-Feng Chiu, and Ling-Chuan Tai

淡江大學資訊工程研究所
Multimedia Information Network MINE. Lab.
Department of Computer Science and Information Engineering of Tamkang University
Email : tshih@cs.tku.edu.tw

摘要

我們發展了一個可支援多媒體資料重用的物件導向多媒體資料庫。此資料庫乃是以物件導向的觀念，定義了四種物件：視框、視框群、資源及資源群。並將資料庫分成兩個階層：視框層和資源層，在視框層中，由視框所組成之視框群乃是一可重複使用的基本單元，而在資源層中一個資源群則是由一群資源所組成，用以供視框及視框群使用。另本資料庫提供四種鏈結，用以建立視框與資源之間的各種關係，並藉以組合成視框群及資源群，以供簡報設計者再次使用。相信，藉由與圖形式使用者介面及先前完成之簡報設計系統的結合，使用者可經由資料庫瀏覽編輯器，直接選取多媒體資源及簡報資料，進而簡單又快速地設計出更精彩、更完美的多媒體簡報。

關鍵字:多媒體資料庫,多媒體簡報,多媒體資料庫管理系統

Abstract

In this paper, we develop a multimedia database system that supports the reuse of multimedia presentations and resources. By the concepts of object-orientation, We define four types of objects: frame, frame group, resource, and resource group. The database consists of two layers: the frame layer and the resource layer. The integration of the database graphical user interface and the Intelligent Multimedia Presentation System will allow users to design a wonderful presentation quickly and easily by searching and selecting the resource and presentation pieces using a database browser.

Keywords : Multimedia Database, Multimedia Presentation, MDBMS, IMMPS*

1. Introduction

Due to the impressive sound and visual effect, and the low price of personal computer hardware, multimedia presentations have been used in many areas. People use multimedia facilities to present their products, plans, and lectures. How to quickly design a multimedia presentation of a high quality has became a hot research issue. Although there are some commercial presentation design software available on the market. However, most presentations generated by these systems communicate with the addresses in a single direction manner. That is, the presentation software does not listen to the listeners' response. For this reason, we have developed an Intelligent Multimedia Presentation System (IMMPS) [7] to overcome the shortage.

Figure 1 shows the three subsystem of the IMMPS: The user interface subsystem, the Database subsystem, and the runtime subsystem. The user interface subsystem provides designer some tools to design his/her presentation, and user can select and retrieval multimedia resources/presentations via the database subsystem . The presentation will work according designer's discipline. In order to correct presentation flow, there is a specification language for describing all behavior of the presentation. The system uses a small rule-based knowledge base to store all rules of the presentation and the fact of user's response. The runtime subsystem will show the multimedia presentation according to the interaction of user, which inferences the presentation knowledge by prolog language .This paper focuses on the discussion of the database subsystem of the IMMPS project, which supports the reuse of multimedia presentations and resources.

The rest of this paper is organized as the follows. In Section 2 presents the Architecture of our database. Some important properties of multimedia resources and presentations are addressed in section 3. The

*This project is supported by the NSC grant : #NSC 86-2213-E032-009 of the Republic of China.

detail of implementation are given in section 4. Finally, a short conclusion and the discussion of our future works are given in section 5.

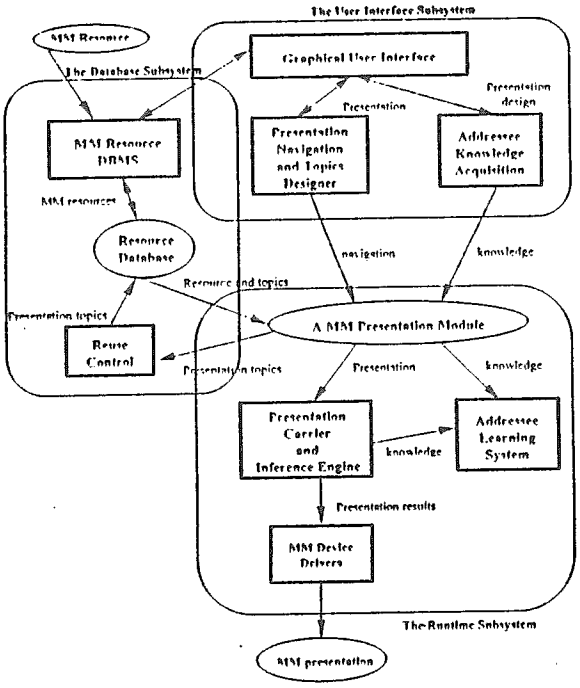

Figure 1: A system for intelligent multimedia presentation designs

2. The Database Architecture

The architecture of the proposed multimedia database consists of two layers: the frame object layer and the resource object layer. Figure 2 gives an overall view of the database. In the architecture, object are connected by links. Two type of objects are implemented: frames and resources. And four types of links are used, as discussed in the following definitions:

Definition: A ***frame*** is denoted by a box which represents related issues that a presenter wants to illustrate.

Definition: A ***resource*** is denoted by a circle with its associated properties given in an attached rounded box, which is a picture, a text description, a video, or other material that can be used in a multimedia computer.

Definition: An ***inheritance link***, denoted by a thick straight line, represents a property inheritance between two frames which is used in the process of knowledge collection.

Inheritance links are used in the process of knowledge collection of an activated frame before the logical inference of the frame proceeds [7]. A message, with or without parameters, show the usage relation between two frames.

Definition: A ***usage link***, represented by a labeled thin straight line, is a link which represents a message passed between two frames.

Definition: An ***aggregation link***, shown by a leader with a single direction arrow, indicates that a frame is using a resource.

An aggregation link connects frames and resources in the two layers of the database. An association may exist in between two resources, For example, an animation resource is associated with a MIDI resource which is used as the background music.

Definition: An ***association link***, is represented by a bi-direction straight curve line between two resources which are corelated.

As well as we have discussed above, we based on the concept of object groups for the reuse of multimedia resources and presentations. An object group is the collection of objects which serves as the basic unit in a presentation. For example, a piece of presentation showing the history of computer consists of several frames associated with a number of resources used in the frame is defined as a reused object. This piece of presentation can be reused in several computer related presentations. Two types of groups are defined in the database: the frame group and the resource group.(both are denoted by circled in figure2).

In the first layer of our database, a frame group is a collection of frames, which is the basic presentation unit to be reused. A resource group in the second layer is a collection of resources to be used by a frame, or a frame group. A frame group or a resource group could become a reusable object stored in the database, if the group is declared by the user via our graphical user interface. After the declaration, an object group becomes an object class. These frame object class and resource class will be reused when instantiated later on. Finally, we have last object in our database named Project, which is a complete multimedia presentation.

In this section, we have discussed the overall picture of our database system. In the next section, we summarize important properties of two types of objects and object groups in our database.

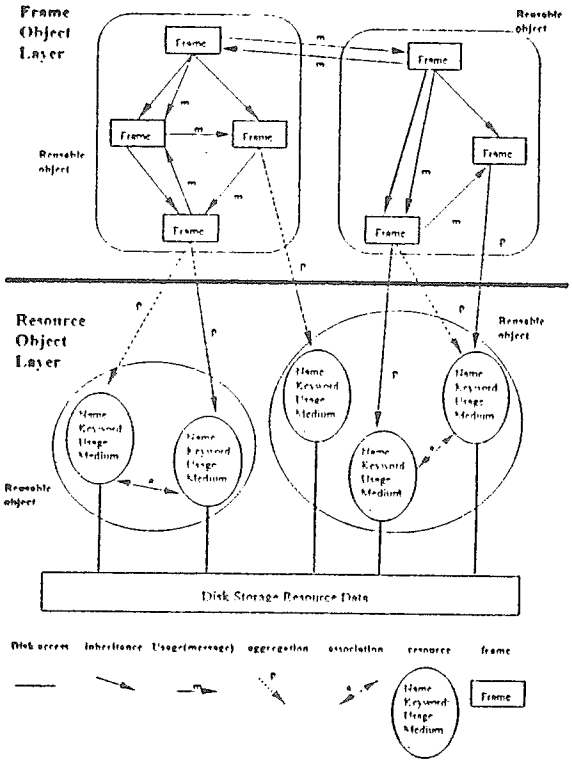

Figure 2 : The database architecture consists of two layers

3. Multimedia Object Attributes

To create a high quality multimedia presentation, not only a good presentation design environment is essential, but good multimedia resource are the key. In order to search the resources conveniently, we have associated with a number of attributes for the resources. We consider the following attributes for objects in the resource object layer of our database:

- **name**: a unique name of the resource.
- **keyword**: one or more keywords are used as the description of a multimedia resource. For instance, name if the city is a keyword of the bitmapped picture of Paris.
- **usage**: how the resource in used (e.g., background, navigation, or focus).
- **medium**: what multimedia device is used to carry out this resource (e.g., sound, video, MPEG, or picture).
- **model**: how the resource in presented (e.g., table, map, chart, or spoken language).
- **temporal** endurance: how long does the resource last in a presentation (e.g., 20 seconds or permanent).
- **synchronization tolerance**: how does a participant feel about the synchronization delay of a resource. For instance, a user usually expects the immediate response after pushing a button for the next page of text. But, the user might be able to tolerate for a video play being delayed for two seconds.
- **detectability**: how does the resource attract a listener (e.g., high, medium, or low).
- **startup delay**: the duration between a message is issued and the corresponding resource is presented, especially wen the resource is on a remote computer connected via network.

- **hardware limitation**: what kind of hardware is essential for carrying out the resource (e.g., MPC level 1, MPC level 2, MPC level 3, or other limitation).
- **version**: the version of this resource file.
- **date**: the date and time this file is created
- **resolution**: the resource of this resource file, specially by X*Y (or 0*0) screen units.
- **start/end time**: for non-permanent resources, the starting cycle and the ending cycle of the piece of video, sound, or other resource that can be used, especially used as a presentation resource. A cycle can be a second, one-ten of second, or a frame number of video/animation.
- **resource descriptor**: a logical descriptor to a physical resource data segment on the disk.
- **association links**: pointer to other resources who have the coexistence relation with the current resource.

Since each resource has a number of attributes, if each query needs to contain all of these attributes, it will be tiresome. Thus, we propose an intelligent mechanism that makes a query easier. The database is built in with a number of inference rules. Each rule describes an if-then relation between two attributes. The following are some of the rules used in our database:

If usage = focus then detectabiliety = high
If model = illustration then medium = picture
If medium= picture then temporal_endurance = permanent
If medium = MPEG then hardware_limitation = MPEG_card
If model = map then medium = picture
If ... etc.

some unspecified attributes can be deduced from others. Thus, a user does not need to specify all

attributes of resource while he/she is using a query to search for the resource.

In the frame object layer, each frame is also associated with a number of attributes:

- **name**: a unique name of the frame.
- **keyword**: one or more keywords are used as the description of a frame:
- **inheritance links**: pointers to other frames which inherit properties from the current frame.
- **usage links**: messages from the current frame to the destination frames, including possible parameters.
- **aggregation links**: pointers to resources which are used in the current frame.
- **presentation knowledge**: logic facts, rules, and a query used in the frame when the frame is open. Note that presentation properties can be presented as logic facts.
- **frame layout**: screen coordinates of resources.

Not only the resource and the frame can be reused, but also the frame group and the resource groups can be reused. In order to reuse a set of resources easily, each resource group is also associated with a number of attributes:

- **name**: a unique name of the resource group.
- **keyword**: one or more keywords are used as the description of a resource group.
- **resources**: recording the resources of the resources group.
- **association links**: the association links in the resource group, which is divided into two types: *internal*: notes the association links between all resources, which is the member of the resource group. *external*: notes the association links between the resources inside the resource group and the resources outside the resource group.

we associated a number of attributes with the frame group, too. each frame group is associated with following attributes:

- **name**: a unique name of the frame group.
- **keyword**: one or more keywords are used as the description of a frame group.
- **frames**: recording the frames of the frame group.
- **inheritance links**: the inheritance links in the frame group, which is divided into two types: *internal*: notes the inheritance links between all frames, which is the member of the frame group. *external*: notes the inheritance links between the frames inside the frame group and the frames outside the frame group.

The above attributes in the two object layers are used in database to retrieve suitable objects/object groups for a multimedia presentation. In the next section, we discuss the implementation of our database.

4 Database Implementation

We implement a graphical user interface (GUI) which allows users to communicate with the MDBMS easily by using Microsoft Visual Basic (VB). This interface is also called "the MDBMS Browser/Editor"(show in figure 3). Based on this interface, user can query and retrieval multimedia resources and presentations easily and quickly, and we use Microsoft Visual C++ (VC) language to implement the sever of our database system, which provides a set of Application Program Interface (API) functions available as Dynamic Linked Libraries (DLLs). The API functions support the construction of the database hierarchy, and allows user to create and edit multimedia objects via the MDBMS Browser/Editor. Based on VB and VC, we develop our multimedia database.

We defined five classes for the frame, resource, frame group, resource group and project objects by C++ language. Properties of all objects are attributes of classes, and the member functions of classes is the operations of objects. Four type of links were stored as the data member of the objects. When the user set up a inheritance link between two frames via our graphical user interface, the record address of destination frame will store in the data member of the resource frame named IL_parents., similarly the address of resource frame will store in the data member of the destination frame named IL_children of. Usage links (for message passing) are created at the runtime as navigation results. An usage link occupies no storage. only the messages are stored as data members of a frame object class in the database. Similarly, association links are implement by stored the address of corresponding resource in each of resource objects. The aggregation link are implement by store the address of the resource and frame in the data member of current frame and resource alike.

As discussed in section 2. The reuse of objects in our system is based on the concept of object group. If the group is declared by the user via graphical user interface, the object group becomes an object class which we defined in the system. Before an object group is stored as an object class in the database, some links be discarded while some links will be maintained. For the frame in a frame group,

the usage (or inheritance) links are divided into two parts: the internal usage (or inheritance) links and the external usage (or inheritance) links. An internal usage (or inheritance) link has its source and destination frames both belong to the frame group. An external usage (or inheritance) link, on the other hand, has its destination frame outside the frame group. When a frame group becomes a frame object class, our system will discard all external links of the frame group. The similar concept is applied to a resource group. When a resource group becomes a resource object class, our system keeps the internal association links and discards all external association links.

The process of instantiation required the allocation of memory or disk storage for the new instantiated object group, and the declaration of external links to other objects in the presentation. For a frame group instantiation, the external usage links and the external inheritance links will be restored by the user. For a resource group instantiation , the external association links need to be given. An instantiation process creates a new object group with new links connected to other objects in the presentation. But, the instantiation process does not duplicate information that can be shared among instance, such as frame layouts or the actual resource data stored on the disk.

5. Conclusion

In this paper, we develop an object-oriented multimedia database. The database sever is written in Visual C++ and Visual Basic run under MS Windows 95/3.1. The database system is to support the design of intelligent multimedia presentations constructed by using our presentation system. User can reuse presentations pieces by the concept of frame group which as the basic reusable unit of a multimedia presentation that generating by the frame object class. Similarly, the generalization of a resource group, or an individual resource as a group, can be reused. The database hierarchy consists of two logical layers for presentation frames and multimedia resources. A number of database commands and their corresponding API (Application Program Interface) functions are designed. System use these API functions to communicate with the graphical user interface called "the MDBMS Browser/Editor " for satisfying the requirement of the system user. By the graphical user interface, system allows a user to organize his/her multimedia resources and presentations pieces easily and quickly. Another article [8] also propose this layered approach for multimedia data modeling. However, the reuse of multimedia objects are not fully discussed. Our approach focus not only on the hierarchy of the database, but the mechanism of reusing multimedia objects.

We are looking at the semantic modeling of multimedia presentation, which the temporal properties of multimedia resources and presentation need to be considered in the semantic model. Another area of our future work is to investigate the content-based retrieval and database indexing techniques for the fast retrieval of multimedia data.

The contributions of this paper are , firstly, we propose a database hierarchy uses an object-oriented approach. This architecture is implemented by Visual C++ and Visual Basic to support our intelligent multimedia presentation designs. Finally, the reuse mechanism allows the user of our system to organize and reuse their presentation easily.

References

[1] Timothy K. Shih, "A Database for Intelligent Multimedia Presentations (An Invited Paper)", in Proceedings of the 8th International Conference on Systems Research, Informatics and Cybernetics, Baden-Baden, Germany, August 14 - 18, 1996.

[2] Timothy K. Shih and Yule-Chyun Lin, "Formal Specification of Multimedia Database Modeling with Z Notations", accepted for publication in Proceedings of the 1996 International Computer Symposium, Taiwan, R.O.C., 1996.

[3] Timothy K. Shih and Chin-Hwa Kuo and Kuna-Shen An, "An Object-Oriented Database for Intelligent Multimedia Presentations", Proceedings of the IEEE International Conference on System, Man, and Cybernetics Information, Intelligence and Systems Conference, 1996.

[4] Timothy K. Shih and Chin-Hwa Kuo, "Database Support for Intelligent Tutoring Software", in Proceedings of the IEEE Second International Conferenceon on Multimedia in Education (M2E2), 1996.

[5] Chi-Cheng Lin,Jiangxu Xiang, "Transformation and exchange of multimedia objects in distributed multimedia systems", Multimedia Systems, Springer-Verlag, pp 12—29, 1996.

[6] Timothy K. Shih, Chin-Hwa Kuo, and Kuan-Shen An, "Multimedia Presentation Designs with Database Support", in Proceedings of the NCS'95 conference, Taiwan, 1995.

[7] Chi-Ming Chung and Timothy K. Shih and Jiung-Yao Huang and Ying-Hong Wang and Tsu-Feng Kuo, "An Object-Oriented Approach and System for Intelligent Multimedia Presentation Designs", Proceedings of the ICMCS'95 conference,

pp 278—281, 1995.
[8] Gerhard A. Schloss and Michael J. Wynblatt, "Presentation Layer Primitives for the Layered Multimedia Data Model", Proceedings of the IEEE 1995 International Conference on Multimedia Computing and Systems, May 15-18, Washington DC, pp 231—238, 1995.
[9] Roger Chen and Dikran S. Meliksetian and Martin Cheng-Sheng Chang and Larry J. Liu, "Design of a multimedia object-oriented DBMS", Multimedia Systems, Springer-Verlag, Vol. 3, pp 217—227, 1995.
[10] Tamer Ozsu and Duane Szafron and Ghada El-Medani and Chiradeep Vittal, "An object-oriented multimedia database system for a news-on-demand application", Multimedia Systems, Springer-Verlag, Vol. 3, pp 182—203, 1995.
[11] ObjectPro/ODB reference manual, Institute for Information Industry, Taiwan,1995.
[12] Shankar Moni and R. L. Kashyap, "A multiresolution representation acheme for multimedia databases", Multimedia Systems, Springer-Verlag, Vol. 3, pp 228—237, 1995.
[13] Michael Lohr and Thomas C. Rakow, "Audio support for an object-oriented database-management system", Multimedia Systems, Springer-Verlag, Vol. 3, pp 286—297, 1995.
[14] Niki Pissinou and Vijay Raghavan and Kanonkluk Vanapipat, "RIMM: A Reactive Integration Multidatabase Model", Informatica, Vol. 19, pp 177—193, 1995.
[15] Dimitris N. Choratas, "Multimedia and the Corporate Database in Intelligent Multimedia Databases", PTR Prentice Hall, chapter 17, pp 305—315, 1994.
[16] Atsuo Yoshitaka and Setsuko Kishida and Masahito Hirakawa and Tadao Ichikawa, "Knowledge-Assisted Content-Based Retrieval for Multimedia Database", IEEE Multimedia Magazine, Winter, pp 12—21,1994.
[17] Jer-Wen Huang, "Multibase: A Heterogeneous Multidatabase Management System", Proceedings of the 18th Annual Interactional Computer Software and Application Conference(COMPSAC 94), IEEE Computer Society Press, pp 332—339, 1994.
[18] Te-Chih Chen and Wei-Po Lin and Chin-An Wu and Chih-Shen Shen, "A Client-Server Database Environment for Supporting Multimedia Applications", Proceedings of the 18th IEEE annual international computer software and application conference (COMPSAC'94), Taipei, Taiwan, pp 215-220, 1994.
[19] Rei Hamakawa,Jun Rekimoto, "Object composition and playback models for handling multimedia data", Multimedia Systems, Springer-Verlag, pp 26—35, 1994.
[20] Keh-Feng Lin and Chueh-Wei Chang and Suh-Yin Lee, "Design of an Interactive Video Database", Proceedings of the 1994 HD-Media Tech. and Application workshop, Taipei, Taiwan, pp PO2-17--PO2-22,1994.
[21] Raymond Paul and M. Farrukh Khan and Ashfaq Khokhar and Arif Ghafoor, "Issues in Database Management of Multimedia Information", Proceedings of the 18th IEEE annual international computer software and application conference (COMPSAC'94), Taipei, Taiwan, pp 209—214, 1994.
[22] Eitetsu Oomoto and Katsumi Tanaka, "OVID: Design and Implementation of a Video-Object Database System", IEEE Transactions on Knowledge and Data Engineering, Vol. 5, No. 4, pp 629—643, 1993.
[23] Matthias Rhiner and Peter Stucki, "Database Requirements for Multimedia Applications", Eurographic Seminars, Tutorials and Perspectives in Computer Graphics", edited by L. Kjelldahl, Multimedia System, Interaction and Applications, pp 269—281, 1991.