Database Support for Intelligent Tutoring Software

Timothy K. Shih and Chin-Hwa Kuo
Department of Computer Science and Information Engineering
Tamkang University
Tamsui, Taiwan, R.O.C.

Abstract

Most multimedia presentations produced by many commercial software do not consider the audience's individual background. IMMPS is a multimedia authoring and database system that allows a presenter to plan the audience's reaction in advance. While the audience is watching a presentation, the underlying inference system is learning from his/her response. This mechanism makes a presentation to be proceeded again act according to the audience's background and knowledge. Thus, the resulting presentation is more diversified.

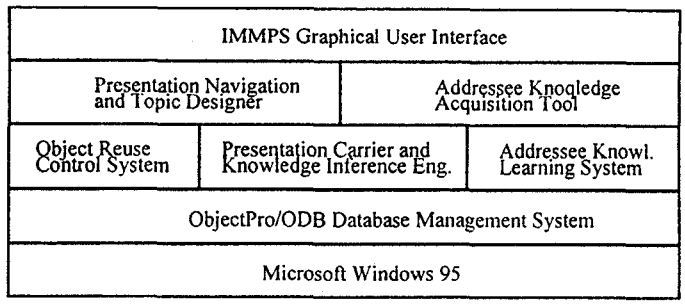

Figure 1: The software architecture of IMMPS

1 INTRODUCTION

As the popularity of multimedia computers increases recently, many multimedia CD ROM titles were also launched by software venders. A number of software systems for producing multimedia software were also available. However, most CD ROM titles generated by these system allow only pre-defined contents, such as video files, a sequence of slide show, or a piece of music, to be presented to the audience. These type of presentations usually communicate with the audiences in a single direction manner, or provide a limited way to listen to the audiences via push buttons or pull-down menus. Not many CD ROM titles support individual tutoring. As a result, many users of the same CD ROM title listen to the same presentation over and over again even though different users mostly have different background and learning speed. For this reason, we propose an Intelligent MultiMedia Presentation System (IMMPS) that allows a presentation designer to construct his/her intelligent presentation as a CD ROM title.

The main contribution of our research is in its delivery of a system which provides: 1.) an intelligent presentation specification language, 2.) an addressee characteristics specification and learning environment, and 3.) a multimedia resources and presentation DBMS with reuse controls. Presentation intelligence is represented by a canonical rule-based format. These knowledge not only include the addressee's background (i.e., common sense of the person who watches the presentation), but allow human reactions to be learned by the presentation program. A database management system is also designed for the CD ROM title designers to organize and store his/her multimedia resource/presentation. This database is associated with a presentation reuse control interface. Presentations can be reused partially or fully upto the designer's choice. Our system supports personalization. Not only the graphical user interface of the generated presentation can be fully customized, but the underlying knowledge of the addressee can be easily updated.

Fig. 1 illustrates the software architecture of our system. The graphical user interface of IMMPS is supported by two sub-systems: the presentation navigation and topic designer, and the addressee knowledge acquisition tool. The first sub-system allows the designer to drag and drop presentation topics, such as picture boxes or music boxes, in a presentation window. Push buttons and hot spot areas are also allowed for the addressee to control presentation navigation. The second sub-system allows the presentation designer to enter knowledge represented as facts and rules similar to the one used in Prolog. While the designer is running his/her presentation, the presentation carrier and knowledge inference engine carry out the multimedia resources, control the sequence of navigation, and perform the knowledge inference. Some of the inference results may assert/retract knowledge facts or rules into/from the presentation. Thus, the presentation program is updating itself as the communication proceeds between the addressee and the computer. The presentation can be reused, which is controlled by our object reuse control system. The system allows two type of reuses: the reuse of multimedia resources and the reuse of multimedia presentations. These objects are stored in our multimedia database which is built on top of a commercial object-oriented database (i.e., Ob-

0-7803-3173-7/96/$5.00

jectPro/ODB). Finally, the first version of our system is run under the MS Windows 95.

We start our discussion with the analysis of some commercial products, and talk about some related works. Then, we present our approach which looks at a presentation from two perspectives. An object-oriented multimedia database is then introduced to support our presentation design. Finally, our conclusions are given in the last section.

2 RELATED WORKS

A number of researchers developed domain specific presentations using artificial intelligence techniques. For example, COMET (COordinated Multimedia Explanation Testbed) [6] uses a knowledge base and AI techniques to generate coordinated, interactive explanations with text and graphics that illustrates how to repair a military radio receiver-transmitter. WIP [1] is able to generate knowledge-based presentations that explain to a user how to use an espresso machine. The work described in [2] integrates knowledge representation systems and a propositional logic theorem prover to create text and map based illustrations showing the situations and plans of a Navy's fleet. APT (A Presentation Tool) [8] automatically generates graphical presentations of relational information. A Piano tutor described in [5] is able to use coordinated media, video, voice, and graphics display, to teach beginners how to play the piano. To support the creation of good multimedia presentations, many articles suggest that a multimedia database supporting fast indexing and synchronization is essential [4]. A distributted database supporting the development of multimedia applications is found in [4]. A mechanism for formal specification and modeling of multimedia object composition is found in [7]. The work discussed in [7] also consider the temporal properties of multimedia resources. The layered multimedia data modeling mechanism [10] suggests a good mechanism to manage multimedia data.

The related works addressed above are mostly academic researches. On the other hand, we also looked at the following commercial products related to multimedia authoring or presentation designs:

1. Authorware Professional by Macromedia, Inc.
2. Director by Macromedia, Inc.
3. Multimedia Viewer by Microsoft
4. Multimedia Toolbook by Asymetrix Corporation
5. Hypermedia System by ITRI (Taiwan)
6. Action! by Macromedia, Inc.
7. Audio Visual Connection by IBM

Authorware uses an event control flow diagram allowing the presenter to specify presentation objects and

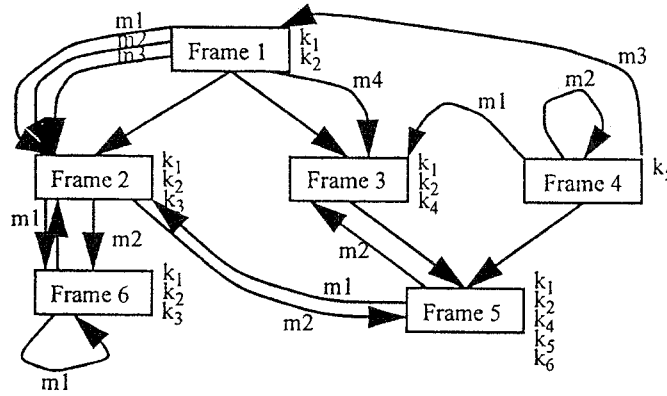

Figure 2: Graph and DAG representations of a multimedia presentation

controls, which can be decomposed into several levels in a hierarchy structure. The system also provides a simple script language for calculation and data manipulation. Other systems (i.e., 3, 4, and 7 above) also provide script languages and API (application program interface) functions. Hypermedia System, Action!, and Director use a time line table allowing actions or objects to be dropped in a particular time slot. Most systems allow users to cut and paste presentation objects or actions via button click and drawing. Multimedia Viewer also provides a set of medium editing tools. Presentation objects produced by these tools can be linked together by a script language supporting functions, data structures, and commands. None of the above systems, however, allows the interactive sequences provided by a user to be learned by a presentation. Our approach, on the other hand, solves this problem.

3 PRESENTATION MODEL

To design a multimedia presentation system, the first thing is to understand what is a multimedia presentation. Most presentations produced by multimedia authoring tools are designed as a hypermedia document. Unlike traditional presentations using slides, multimedia presentations are proceeded in a nonlinear manner in that push buttons are provided for the user to navigate through different related issues. These presentation issues are not totally independent for two reasons. Firstly, two issues closely related to each other should be linked such that the audience watches the presentation can refer to related issues easier by following the links. For instance, a presentation touring Paris may links a picture of the Paris Tower with a description, either by audio or text, of its history. Secondly, two issues presented in a presentation may share common information. There is no reason of storing duplicated data in one presentation, which makes a consistent update becomes tiresome. For instance, the addressee's name and

background used in an interactive presentation should be stored only once in the presentation. While updated, the addressee's name will be changed consistently. We suggest that a presentation can be designed from two different perspectives: the *navigation view* and the *representation view*. From the navigation view, a presentation is a graph with nodes as presentation windows and edges as navigation links. From the representation view, information that can be shared among windows are background of the audiences (e.g., the addressee's name or knowledge to the presentation topic), multimedia resources (e.g., a text document file or a video file showing a mechanical operation), or other knowledge useful in the presentation. A property inheritance structure such as a tree or a DAG (Directed Acyclic Graph) is suitable for our knowledge representation architecture. Fig. 2 illustrates these two structures. In the figure, a *frame* (i.e., a presentation window) is a composed object which represents a topic that a presenter wants to discuss. A frame may contain push buttons, one or more multimedia resources to be presented, and a number of knowledge rules (e.g., k1, k2, k3). A *message* (e.g., m1, m2) with optional parameters is passed between two frames (or back to the same frame). The graph edges represent navigation links are shown in curve lines, with message names as labels. The DAG edges represent knowledge inheritance are shown in straight lines without label. In the figure, to the right of each frame, we show the knowledge rules that can be used in the frame. Even knowledge rules "k1", and "k2" are shared among Frame 1, Frame 2, and Frame 3, they are stored only once in Frame 1. Note that multiple inheritance is also allowed, as Frame 5 inherits knowledge rules from both Frame 3 and Frame 4.

There are a number of restrictions applied to our message passing system and knowledge inheritance system. For instance, a message passed in between two frames has a unique name. And, only the destination frame can receive the specific message sent to it. Each message has only one source and one destination. A child frame inherits all knowledge rules from its parent frames. The relation of knowledge inheritance is transitive. However, the inheritance architecture is acyclic. That is, a frame can not be a parent frame of its own. If a child frame contains a knowledge rule that has the same rule name as one of the rules the child frame inherites (directly or indirectly), the rule defined in the child frame overrides the one from its parent frames. If two rules belong to two frames have the same name, the rule should be stored in a common parent frame only once to avoid inconsistency.

4 THE DATABASE

To design a good multimedia presentation, not only a good software system, but high quality multimedia resources are essentially important. Also, to increase the efficiency of design, reuse of pre-designed presentation pieces is a key. Not many presentation design system has an underlying database management system to support presentation reuse. In our research, the development of a multimedia database is considered from the beginning. We survey a number of techniques to develope multimedia databases. The strategies of storing multimedia resources has four approaches:

- relies on a regular file system
- uses a traditional database manegement system (e.g., relational DBMS), with the support of an object-oriented interface
- uses an object-oriented database management system, with user interface support
- bases on object-oriented concepts, design the database from scratch

The first approach relies on the users to manage multimedia resources by themself. There is no support of reuse of presentations. Most presentation systems allow cut and paste of portions of a presentation. However, this is not an ideal strategy in general due to the limitation of a file system, such as the inflexiblity of object composition and sharing. The second approach relies on a relational DBMS. However, due to the natural of multimedia presentations, it is easier to organize a presentation using an object-oriented methodology [3, 9]. The difference between a traditional DBMS and an object-oriented DBMS makes the usage of a relational-based multimedia database system becomes inefficient [9]. Thus, the third approach overcomes the second by using an object-oriented DBMS as the underlying system. Even the underlying system may not be designed specially for multimedia data, we found that most object-oriented DBMSs provide the binary data type, which is useful for storing pictures, sound, video, etc. The last approach is the most efficient way in general. However, it is quite time consuming to design this type of multimedia database. For the above reasons, we take the third approach and use a commercial object-oriented DBMS. In the following sub-sections, we discuss the development of our multimedia database supports the reuse of multimedia presentations.

4.1 The Database Hierarchy

The advantages of an object-oriented database over a traditional database are in its data sharing via property inheritance, and in the convenience of object reuses. With the helps of these techniques, our database management system supports the integration of multimedia resources and presentation designs. The hierarchy of the proposed multimedia database consists two layers: the *frame object layer* and the *resource object layer*. Fig. 3 gives an overall view of the database. In the hierarchy, objects are connected by links. Two types

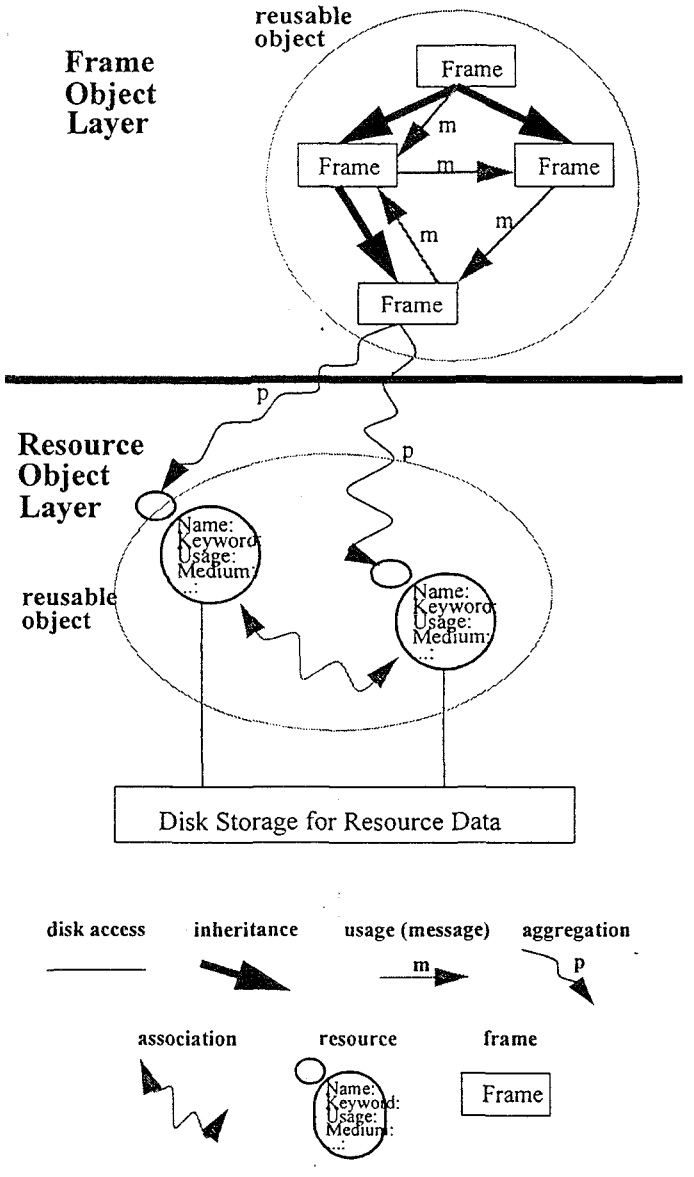

Figure 3: The database hierarchy consists of two layers

of objects are implemented: frames and resources. And four types of links are used, as discussed in the following definitions:

Definition: An *inheritance link*, denoted by a thick straight line, represents a property inheritance between two frames in fig. 3. ■

Inheritance links are used in the process of knowledge collection of an activated frame before the logical inference of the frame proceeds. A message, with or without parameters, shows 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 curve line 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 arrowed curve line between two resources which are corelated. ■

The two types of objects in our database are frames and resources. A frame is denoted by a box while a resource is represented by a circle with its associated properties given in an attached rounded box. The actual data of a resource is stored in the commercial multimedia file format on the hard disk.

So far, we have discussed the overall picture of our database system. In the next section, we summarize important properties of the two types of objects in our database.

4.2 Multimedia Object Attributes

Multimedia resources are recorded or captured via camera, tape recorder, or video camera, converted to their digital formats, and saved on the disk. These resource files can be reused in different presentations. A resource is associated with a number of attributes. We consider the following attributes for multimedia resources 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 of the city is a keyword of the bitmapped picture of Paris.
- **usage**: how the resource is 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 is 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 when 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, level 2, level 3, or other limitations).
- **version**: the version of this resource file.
- **date**: the date and time this resource file is created.
- **resolution**: the resolution of this resource file, specified by $X \times Y$ (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 resources that can be used, especially used as a presentation resource. A cycle can be a second, one-tenth of a second, or a frame number of a video/animation.
- **resource descriptor**: a logical descriptor to a physical resource data segment on the disk.
- **association links**: pointers to other resources who have the coexistence relation with the current resource.

Since each resource has a number of attributes, if each query searching a resource 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 detectability=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 un-specified attributes can be deduced from others. Thus, a user does not need to specify all attributes of a 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 the frame.
- **inheritance links**: pointers to other frames which inherit properties from the source frame.
- **usage links**: messages from the source frame to the destination frames, including possible parameters.
- **aggregation links**: pointers to resources which are used in the source frame.
- **presentation knowledge**: logic facts, rules, and a query used in the frame when the frame is open. Note that presentation properties can be represented as logic facts.
- **frame layouts**: screen coordinates of resources.

Presentation knowledge and frame layouts are provided by the user via our graphical user interface. In the next section, we discuss the reuse of frames and resources.

4.3 Object Reuse

The reuse of objects in our system is based on the concept of *object groups*. An object group is a collection of objects which serves as the basic unit in a presentation. For example, a piece of presentation showing the history of computers consists of several frames associated with a number of resources used in the frames is defined as a reusable object. This piece of presentation can be reused in several computer science related lectures. Two types of groups are defined in the database: the *frame groups* and the *resource groups* (both are denoted by circles in fig. 3). 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 object class* will be reused when instantiated later on.

Before an object group is stored as an object class in the database, some links will be discarded while some links will be maintained. For the frames 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 keeps the internal inheritance links, the internal usage links, and the aggregation links. But the 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 requires 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 instances, such as frame layouts or the actual resource data stored on the disk. The management of object reuse is assisted by our presentation reuse control interface and a resource browser. The user is able to use our system to organize their presentations and resources systematically.

5 CONCLUSIONS

In this paper, we introduced an intelligent multimedia presentation system allows a presenter to design intelligent presentations. The presentations designed allow addressees' response to be learned via knowledge assertions as knowledge inference side effects of some preplanned knowledge rules. Our new proposed model, by using an object-oriented approach, allows a presenter to design his/her presentation as a hypermedia document with navigation specified as messages among frames. This model, by allowing knowledge inheritance, also facilitates data sharing and ensures a consistent updating of knowledge. A presentation design is entered via our graphical user interface. Different components of a frame are given in different windows. The user interface takes these components and produces a presentation which is run by our presentation carries/inference engine subsystem.

An object-oriented multimedia database is also introduced. The database system is to support the design of intelligent multimedia presentations constructed by using our presentation system. A presentation consists of a number of frame groups. The generalization of a frame group, named a frame object class, serves as the basic reusable unit of a multimedia presentation. Similarly, the generalization of a resource group, or an individual resource as a group, can be reused. This reuse mechanism, supported by a graphical user interface, allows a user to organize his/her multimedia resources and presentation pieces easily.

The intelligent multimedia presentation system as well as its supporting database run under the MS Windows 95. The implementation language is Visual Basic, C++, and Prolog. Some sample applications are also designed to show the usage of our system. These applications are: a student study plan consultant, a classic music browser, and a Chinese violin tutor.

Our contributions in this paper are, firstly, we use object-oriented methodology and artificial intelligence techniques to design a model for presentations by specifying relations between frames. Secondly, a language and a user interface are defined and its supporting environment is implemented. Also, we propose a database hierarchy uses an object-oriented approach. This hierarchy is then implemented on the top of the ObjectPro/ODB database management system to support our intelligent multimedia presentation designs. And, the reuse mechanism allows the user of our system to organize and reuse their presentation easily. Finally, some sample applications are designed showing our research results. This system can be used for general purpose presentations or demonstrations in different fields such as education, training, product demonstration, and others.

References

[1] E. Andre. "WIP: The Automatic Synthesis of Multimodal Presentations". In M. T. Maybury, editor, *In Intelligent Multimedia Interfaces*, pages 75–93. American Association for Artificial Intelligence, 1993.

[2] Y. Arens. "Presentation Design Using an Integrated Knowledge Base". In J. W. Sullivan and S. W. Tyler, editors, *In Intelligent User Interfaces*, pages 241–258. ACM Press, 1991.

[3] C. R. Chen, D. S. Meliksetian, M. C.-S. Chang, and L. J. Liu. "Design of a multimedia object-oriented DBMS". *Multimedia Systems, Springer-Verlag*, 3:217–227, 1995.

[4] T.-C. Chen, W.-P. Lin, C.-A. Wu, and C.-S. Shen. "A Client-Server Database Enviornment for Supporting Multimedia Applications". In *Proceedings of the 18th IEEE annual international computer software and application conference (COMPSAC'94)*, pages 215–220, 1994.

[5] R. B. Dannenberg and R. L. Joseph. "Human Computer Interaction in the Piano Tutor". In M. M. Blattner and R. B. Dannenberg, editors, *In Multimedia Interface Design*, pages 65–78. 1992.

[6] S. K. Feiner and K. R. McKeown. "Automating the Generation of Coordinated Multimedia Explanations". In M. T. Maybury, editor, *In Intelligent Multimedia Interfaces*, pages 117–138. 1993.

[7] T. D. C. Little and A. Ghafoor. "Synchronization and Storage Models for Multimedia Objects". *IEEE Journal on Selected Areas in Communications*, 8(3):413–427, 1990.

[8] J. D. Mackinlay. *"Automatic Design of Graphical Presentations"*. PhD thesis, Stanford University, 1987.

[9] M. T. Ozsu, D. Szafron, G. El-Medani, and C. Vittal. "An object-oriented multimedia database system for a news-on-demand application". *Multimedia Systems, Springer-Verlag*, 3:182–203, 1995.

[10] G. A. Schloss and M. J. Wynblatt. "Presentation Layer Primitives for the Layered Multimedia Data Model". In *Proceedings of the IEEE 1995 International Conference on Multimedia Computing and Systems, May 15-18, Washington DC*, pages 231–238, 1995.