# 行政院國家科學委員會補助專題研究計畫成果報告

※※※※※※※※※※※※※※※※※※※※※※※※※
※　　　　　　　　　　　　　　　　　　　　　　　※
※　　　　多媒體隨選資料庫管理系統　　　　　　　※
※　　　　　　　　　　　　　　　　　　　　　　　※
※　　　　　　　　　　　　　　　　　　　　　　　※
※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：■個別型計畫　　□整合型計畫
計畫編號：NSC　89－2213－E－032－003
執行期間：88 年 08 月 01 日至 89 年 07 月 31 日

計畫主持人：劉虎城　教授

本成果報告包括以下應繳交之附件：
- □赴國外出差或研習心得報告一份
- □赴大陸地區出差或研習心得報告一份
- □出席國際學術會議心得報告及發表之論文各一份
- □國際合作研究計畫國外研究報告書一份

執行單位：淡江大學資訊工程學系

中　華　民　國　89年　09月　26日

# 行政院國家科學委員會專題研究計畫成果報告

## 國科會專題計畫成果報告撰寫格式說明
## Preparation of NSC Project Reports

計畫編號：NSC 89-2213-E-032-003
執行期限：88 年 8 月 1 日至 89 年 7 月 31 日
主持人：劉虎城　　淡江大學資訊工程學系

一、中文摘要

計劃中我們提出一個支援多媒體資料庫的編輯需求系統。本系統的主要目的在於提出在教學環境中支援數種電腦相關課程設計的系統。在系統的軟體架構中包含了數個子元件，其中有編輯和撥放系統，課程管理介面和一個白板系統。利用多媒體資料庫的支援設計者能夠更有效率的設計、組織與重用課程內容。

關鍵詞：多媒體資料庫管理系統、物件重用、編撰需求系統

**Abstract**

We present an instruction on demand system with its underlying multimedia database support. The system is used in the realization of several computer science related courses in our university. Preliminary experience shows that, with the availability of our multimedia database management system, instructors can design, organize, and reuse courses efficiently.

**Keywords**: Multimedia Database Management System, Object Reuse, Instruction on Demand .

### 1. Introduction

Multimedia computing and networking can improve our education facility. With the assistant of multimedia computers and Internet, lectures, assignments, and tests can be stored by instructors and retrieved by students for the purpose of distance education. Through Internet, multimedia presentations are broadcasted as instruction on demand. In order to support the production of multimedia applications, the management of multimedia resources (e.g., video clips, pictures, sound files) is important. For instance, multimedia presentations can be designed as building blocks which can be reused. To facilitate multimedia application design, many articles indicate the need of a multimedia database [1, 5]. A multimedia database is different from a traditional relational database in that the former is object oriented while the latter relies on entity relations. Moreover, a multimedia database needs to support binary resource types of large and variable sizes. Due to the amount of binary information that need to be processed, the performance requirement of a multimedia database is high. Clustering and indexing mechanisms support multimedia databases are thus important. In this paper, a multimedia database architecture for instruction on demand applications is proposed.

The paper organized as the following.. A distributed instruction demand system is discussed in section2. A multimedia database hierarchy supports the instruction system is proposed in section 3. Finally, we give some conclusion in section4.

### 2. System Architecture

Distance learning or so called remote classroom relies on network and multimedia facilities and provides teachers an instruction environment, either via analog or digital transmission, such that the instructor and his/her students can join the class in different locations. On the other hand, with the availability of high performance network, Video on Demand (VOD) systems provide customers to access digital video randomly and remotely. In this paper, we propose a system which combines functions of the above two types of systems, namely an Instruction on Demand System. The proposed system supports instructors to design course materials, including lectures, homework, and tests. Students and the instructor in a class communicate with each other through the help of the system. Our research focuses contain several software levels. Issues include: 1)Instruction on demand applications, 2)Multimedia documentation database, 3)Multimedia information networking system, 4)Disk accessing control mechanism, 5)Quality of service and media synchronization.

In order to realize different levels of research focuses, the software architecture is divided into four levels. The user interface layer in the architecture has a number of applications, including a multimedia lecture design system. Since a multimedia lecture is essentially a multimedia presentation, our lecture design system aims to provide a lecture design and playback facility. An instruction administrator will keep the information of which students are enrolled in the class. The white board subsystem provides an environment for the students and the instructor to discuss either in verbal or text form.

The second layer of the architecture is a multimedia database management system (MDBMS). The purpose of this MDBMS is to support many instruction on demand applications. Lectures, assignments, tests, and other course materials will be stored in the MDBMS via the help of an underlying multimedia networking and disk accessing system. The hierarchy of this MDBMS is to fit the need of instructions, which consists of five layers. An object oriented approach is used to facilitate the design. Therefore, one of the focuses of this MDBMS is on its instruction reusability. While instruction objects of various types are stored in the MDBMS hierarchy, the system also supports a dynamic type checking scheme allowing object composition/decomposition. Type constructors and converters based on the concepts of time and space are also provided.

The third layer is a multimedia information g library, which is based on Message Passing Interface (MPI). MPI is a standard of communication mechanism designed for parallel machines, especially for those incorporated with distributed memory. We aim to provide a solution to the communication facility of the multimedia instruction on demand system. We investigated MPI functions and source code provided by a vendor. Our objective is to realize the communication standard not only on the Unix machine (as is the source code provided), but also on the windowing systems by Microsoft. On the top of MPI functions, we design and implement an encapsulation layer which will be accessed by our multimedia database management system. For the sake of performance efficiency, we also extend MPI functions via enhancing its underlying disk accessing mechanism suitable for our database architecture.

The last layer of our system design is a distributed disk accessing system, which is built on the top of Windows and Unix. In order to achieve Quality of Service, disk scheduling mechanisms of our system relies on the logic block read/write statements of SCSI to access local disk data. Also, disk buffers are used for remote disk access.

A control daemon is designed to control system status, which includes database accessing parameters, communication parameters, synchronization parameters, and disk accessing parameters. The control daemon is installed in the instructor workstation with a status buffer shared by all workstations.

## 3. The Database Hierarchy

In order to support the instruction on demand system, an in-house-designed multimedia database management system is used. The database hierarchy is designed for storing multimedia lectures, student homework, and tests. These multimedia presentations are compound objects consist of various level of materials. The hierarchy of the MDBMS has five levels:1)Database layer, 2)Course layer, 3)Lecture layer, 4)Scene layer, 5)Resource layer.

The MDBMS allows multiple databases. There are two types of databases: the instruction database (i.e., **InstDatabase**) and the class database (i.e., **ClassDatabase**). Each instruction database contains one or more courses. Related courses can be grouped. This course group (i.e., CG), through a declaration, can become a course object class (i.e., COC) in a class database. A COC is a reusable component. It can be instantiated to another course group with alternative properties. The purpose of object reuse include information sharing and prototyping. Information sharing, in this circumstance, means that a COC and other COCs can share the same lecture object class (i.e., LOC). Prototyping means that the internal structure of a course is intact in an instantiation process, before the object group is confirmed to some new properties. In the same way, a course has one or more lectures. A lecture group can be declared as a lecture object class for reuse. A lecture can be a lecture presentation, a homework, or a test. A lecture can be designed in our presentation design interface, or designed by using other commercial multimedia presentation software. A lecture can also be attached with a script. If so, the script keeps a sequence of interactions produced by the user who navigates through the lecture. This lecture can be played back if necessary. A lecture is a collection of windows, called scenes in our system. Scenes can be reused as well. A scene object can inherit properties from other scenes. A scene send out navigation message (issued by the user) to scenes (including the origin scene) in order to control navigation sequences. Similarly, scene object class (i.e., SOC) can be reused. Finally, a scene contains a number of presentation resources. A presentation resource may have a synchronization requirement to another presentation resource. And, two resources may have an co-existence relation, or a similarity relation. The resource layer is the last layer of the database hierarchy. Binary Large OBjects (BLOBs) in this layer can be shared by object groups or object classes. Note that, the object class hierarchy of a class database is different from the one of an instruction database. Course, lecture, and scene object classses can be the top level objects in the hierarchy. In this way, the three types of object classes can be identified for reuse at the top level.

### 3.1 Database Operations

Our database server handles database operations from the user's program by means of application program interface (API) functions. Some of these functions came from traditional database systems, whereas others are new for handling multimedia resources. These database commands are implemented as C ++ classes and methods.

### 3.2 Object Reuse in the Database

When we design our database hierarchy, reusibility is a focus. It is the advantage of an object-oriented database system that allows the reuse of classes. Moreover, the hierarchy of our database allows information sharing of multimedia BLOBs.

There are three levels of objects in the database hierarchy that can be reused: courses, lectures, and scenes. The basic block to be reused is an object group. A singular object can be bound in a group for reuse as well. An object group, can be declared as an object class. The object class can be instantiated to another object group for reuse.

The declaration of object class involves a creation of database space for the new class. Since a course group uses a number of lecture groups, the declaration of a course object class enforces the declarations of the used lecture object classes as well. The instantiation process is the reverse of a declaration process. In general, we treat object groups and object classes as separated entities in the database so that a modification of one will not affect the content of the other. Object reuse in the lecture layer is similar to the one in the course layer. However, reuse in the scene layer is different. Since a scene is essentially a window, the navigation among windows are implemented by message passing. Moreover, a scene can inherit information from another scene by means of an inheritance link. Therefore, the declaration and instantiation of a scene object class and a scene group need to manipulate usage links and inheritance links. These two types of links can be internal or external. An internal link has its source and destination scenes both lay within an object group (or an object class). An external link of an object group or class has either the source or the destination of the link outside the object group or class.

Whether it is an object group or an object class hierarchy, the last level of objects are multimedia resources. Resources are shared among groups and classes, or between a group and a class. The update of a resource will change the multimedia instruction content of the entire database.

We implement the database server on a distributed environment. The concurrent access control is centralized by means of a locking mechanism. In an object-oriented database, locking depends on the granularity of the objects in an object hierarchy in two ways: object composition and class inheritance. Object composition needs to maintain the locking of objects at different levels so that locking on a component still leave a free space for the containing object. But, locking on a compound object at a higher level enforces locking on its descendent objects. Similarly, in a class hierarchy, when a user is about to change a parent class, its child classes can not be changed by another user. Therefore we have defined an object locking compatibility table. In general, if a container has a read lock by a user, its components (and itself) can have the read access by another user, but not the write access. However, the parent objects of the container can have both read and write access by another user.

## 4 Conclusions

In this paper, we discussed research issues of distributed multimedia databases. We also proposed a MDBMS to support our instruction on demand system. We implement our system on Windows/95. We hope that, the proposed database hierarchy will ease the design and reuse of multimedia presentations.

## References

[1] Te-Chih Chen, Wei-Po Lin, Chin-An Wu, and Chih-Shen Shen, "A Client-Server Database Environmemt for Supporting Multimedia Applications,"in proceedings of the 18th IEEEannual international computer software and application conference(COMPSAC'94), Taipei, Taiwan, 1994,pp 215-220.

[2] C.Y. Roger Chen and Dikran S. Meliksetian and Martin Cheng-Sheng Chang and Larry J. Liu, "Design of a multimedia object-oriented DBMS," Multimedia Systems, Vol. 3, 1995, Spring-Verlag, pp 217-227.

[3] Thomas D. C. Little and Arif Ghafoor, "Synchronization and Storage Models for Multimedia Objects," IEEE Journal on Selected Areas in Communications, Vol. 8, No 3, April, 1990, pp 413-427.

[4] M.Tamer Ozsu and Duane Szafron and Ghada ElMedani and Chiradeep Vittal, "An object-oriented multimedia Database system for a news-on-demand application", Multimedia Systems, Vol. 3,1995, Springer-Verlag, pp 182-203.

[5] Raymond Paul, M. Farrukh Khan, Ashfaq Khokhar, and Arif Ghafoor, "Issues in Database Management of Multimedia Information," in proceedings of the 18th IEEE annual international computer software and application conference (COMPSAC'94), Taipei, Taiwan, 1994, pp 209-214.