

## **INFORMATION TO USERS**

**This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.**

**The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.**

**In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.**

**Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.**

**Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.**

# **UMI**

**A Bell & Howell Information Company  
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA  
313/761-4700 800/521-0600**





Université d'Ottawa • University of Ottawa



**Design and Implementation of a  
Broadband Multimedia TeleLearning  
System**

by

***Ruiping WANG***

***Department of Electrical and Computer Engineering***

***University of Ottawa***

**© Ruiping Wang  
*April 12, 1997***



National Library  
of Canada

Acquisitions and  
Bibliographic Services

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

Bibliothèque nationale  
du Canada

Acquisitions et  
services bibliographiques

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file Votre référence*

*Our file Notre référence*

**The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.**

**The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced with the author's permission.**

**L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.**

**L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.**

0-612-20960-1

# **Abstract**

The advance of the networking and the multimedia technologies are making possible a wide range of interactive applications and services to be built which may eventually change the way that people live. TeleLearning is one prospective area among them which has obtained lots of attentions from both the educators and the researchers.

In this thesis we discuss a Multimedia Interactive TeleLearning System under development in the Multimedia Information Research Laboratory at the University of Ottawa. The system aims at providing a seamlessly integrated environment for TeleLearning using the latest telecommunication and multimedia information processing technologies. It provides a new type of learning service- Course On Demand, which makes learning more accessible, more flexible, and with more proper help.

The system basically consists of a media production center, a courseware author site, a courseware database, a courseware user site, and an on-line facilitator site. All these components are distributed over an ATM network and work together to offer a multimedia interactive courseware service. An MHEG-based model is exploited as the system architecture to achieve the real-time, reusable information interchange through heterogeneous platforms. The system architecture, courseware-processing strategies, courseware authoring issues, as well as the implementation of the first prototype are presented in this thesis.

# **Acknowledgment**

I would like to thank my supervisor- Dr. Ahmed Karmouch for his motivation and support. His work ethics and feedback have made this thesis a much better piece of work. In particular, his comments and suggestion have made me a much better report writer and researcher than I was before.

I would also like to thank my family and all the people who have helped me through the hardest time after my family tragedy. Without all of them, who are all around in Ottawa, Toronto, Hong Kong, and Mainland China, this thesis work and the continuing of my life would have been impossible.

In the end, I would like to thank Telecommunication Research Institute of Ontario and TeleLearning National Center of Excellence, for their sponsorship.

# Table of Content

<b>ABSTRACT .....</b>	<b>1</b>
<b>ACKNOWLEDGMENT .....</b>	<b>2</b>
<b>TABLE OF CONTENT.....</b>	<b>3</b>
<b>LIST OF FIGURES.....</b>	<b>6</b>
<b>CHAPTER 1 INTRODUCTION.....</b>	<b>1</b>
1.1 LEARNING: THE PAST AND THE FUTURE.....	1
1.1.1 <i>Traditional Methods of Teaching</i> .....	2
1.1.2 <i>Paradigm Shifts in Learning</i> .....	6
1.2 TERMINOLOGY RELATED TO MULTIMEDIA TELELEARNING .....	8
1.3 CURRENT STATUS OF TELELEARNING .....	10
1.3.1 <i>Broadcasting Model</i> .....	11
1.3.2 <i>CD-ROM/PC Model</i> .....	12
1.3.3 <i>Network-based Model</i> .....	13
1.4 THESIS OUTLINE .....	14
<b>CHAPTER 2 MULTIMEDIA AND HYPERMEDIA- APPLICATIONS AND STANDARDS .....</b>	<b>16</b>
2.1 MULTIMEDIA AND HYPERMEDIA APPLICATIONS.....	17
2.1.1 <i>The Hyper-News at 11:00 Application</i> .....	17
2.1.2 <i>A Digital Library Application</i> .....	18
2.2 STANDARDS AND METHODOLOGIES FOR MULTIMEDIA AND HYPERMEDIA INFORMATION DELIVERY .....	19
2.2.1 <i>HyTime: Hypermedia/Time-based Structuring Language</i> .....	20

2.2.2	<i>MHEG: Multimedia and Hypermedia Information Coding Expert Group</i> .....	28
2.3	COMPARISON OF HYTIME AND MHEG.....	40
2.3.1	<i>Multimedia and Hypermedia Authoring and Publishing</i> .....	41
2.3.2	<i>Real-time Interchange and Presentation</i> .....	42
2.3.3	<i>Interactive Applications</i> .....	43
<b>CHAPTER 3 A BROADBAND MULTIMEDIA TELELEARNING SYSTEM</b> .....		<b>44</b>
3.1	DESIGN RATIONALE OF A BROADBAND MULTIMEDIA TELELEARNING SYSTEM.....	44
3.1.1	<i>Requirement of an Innovative Multimedia TeleLearning System</i> .....	45
3.1.2	<i>Choosing Technologies</i> .....	48
3.2	GENERIC ARCHITECTURE.....	51
3.3	AN MHEG-BASED MODEL FOR INFORMATION INTERCHANGE.....	53
3.4	COURSEWARE PROCESSING: MODELS AND STRATEGIES.....	55
3.4.1	<i>Courseware Production</i> .....	56
3.4.2	<i>Courseware Storage and Retrieval</i> .....	58
3.4.3	<i>Courseware Presentation</i> .....	59
<b>CHAPTER 4 COURSEWARE AUTHORIZING IN MITS</b> .....		<b>60</b>
4.1	MULTIMEDIA COURSEWARE PRODUCTION.....	61
4.1.1	<i>General Process for Multimedia Courseware Production</i> .....	61
4.1.2	<i>Courseware Authoring in MITS</i> .....	62
4.2	TEACHING ARCHITECTURES.....	63
4.3	INTERACTIVE DOCUMENT MODELS.....	66
4.3.1	<i>Static Interaction and Dynamic Interaction</i> .....	67
4.3.2	<i>A Hypermedia Document Model</i> .....	69
4.3.3	<i>An Interactive Multimedia Document Model</i> .....	71
4.4	MHEG OBJECT AUTHORIZING.....	74
4.4.1	<i>An MHEG Class Library</i> .....	75

4.4.2 A Courseware Class Library.....	78
4.5 IMPLEMENTATION ISSUES FOR A COURSEWARE EDITOR IN MITS.....	79
4.5.1 Framework.....	80
4.5.2 Template.....	80
4.5.3 Issues for GUI Design.....	81
<b>CHAPTER 5 IMPLEMENTATION OF A COURSEWARE NAVIGATOR IN A BROADBAND MULTIMEDIA TELELEARNING SYSTEM.....</b>	<b>83</b>
5.1 A BROADBAND MULTIMEDIA TELELEARNING SYSTEM .....	83
5.1.1 The MEDIABASE System.....	84
5.1.2 Fitting a Broadband Multimedia Courseware Delivery System into MEDIABASE .....	86
5.2 DESIGN ISSUES OF A COURSEWARE NAVIGATOR ON WINDOWS 95.....	88
5.2.1 Feature Set Analysis and Design.....	89
5.2.2 Multimedia in Windows 95 .....	91
5.3 APPLICATION DEVELOPMENT .....	93
5.3.1 User Interface Construction .....	95
5.3.2 Client Module Construction.....	95
5.3.3 New Application Classes Design.....	96
5.4 SAMPLE LEARNING SESSIONS .....	97
5.5 SUMMARY AND RELATED WORK .....	103
<b>CHAPTER 6 CONCLUSION.....</b>	<b>104</b>
6.1 SUMMARY .....	104
6.2 SUGGESTIONS FOR FUTURE WORK .....	106
<b>BIBLIOGRAPHY .....</b>	<b>108</b>
<b>PUBLICATION.....</b>	<b>112</b>

# List of Figures

Figure 2.1 HyTime module inter-dependencies .....	22
Figure 2.2 HyTime location address module .....	25
Figure 2.3 HyTime document processing model .....	27
Figure 2.4 MHEG objects life cycle .....	33
Figure 2.5 An example of synchronization defined at application level .....	35
Figure 2.6 An example of atomic synchronization and elementary synchronization .....	36
Figure 2.7 Multimedia/Hypermedia interchange model .....	38
Figure 2.8 MHEG for interchange and identification of multimedia information .....	39
Figure 2.9 Interchange of MHEG objects .....	40
Figure 3.1. A generic architecture for MITS .....	51
Figure 3.2 An MHEG-based model for MITS .....	53
Figure 3.3 Processing model for multimedia courseware .....	55
Figure 3.4 Courseware processing modules in MITS .....	57
Figure 3.5. Client-server model in MITS .....	59
Figure 4.1 General process for multimedia courseware production .....	61
Figure 4.2 Layers for multimedia courseware authoring in MITS .....	62
Figure 4.3 Hypermedia document model .....	70
Figure 4.4 Interactive multimedia document model with an ATM course as an example .....	73
Figure 4.5 A basic MHEG class library .....	75
Figure 4.6 An interactive multimedia courseware class library .....	78
Figure 5.1 A generic architecture of the MEDIABASE system .....	84
Figure 5.2 Interactive multimedia courseware platform .....	87
Figure 5.3 The first screen of the navigator .....	98
Figure 5.4 MIRL TeleSchool registration screen .....	98

**Figure 5.5 The course presentation screen ..... 101**  
**Figure 5.6 The screen for updating the student profile..... 102**  
**Figure 5.7 The screen for browsing the library..... 102**

# Chapter 1

## Introduction

### 1.1 Learning: the Past and the Future

Learning refers to "a relatively permanent change in behavior or knowledge brought about by practice or experience". A piece of information is said to be learnt and becomes one's knowledge when it is understood and memorized [Grace, 95].

Traditionally, knowledge is transferred to learners linearly in the form of textbooks, speech or report [Grace, 95]. This is the result of the technological factors and practical constraints. Consequently, the "classroom-and-book" education model has widely spread since 1900's. The students are grouped according to their ages, and gathered together in a classroom so as to learn a specific topic from the teachers' presentation at a specific time. This education model has been efficient for a long period of time. However, people also see its constraints in some aspects. One of them could be the limited accessibility for those who have problems joining a class at a specific time.

Another constraint would be the high student-to-teacher ratio which has limited the learning efficiency for each individual learner. Furthermore, for training employees in companies of different sizes, just-in-time knowledge is required to be transmitted and students are usually with much different knowledge background. In such cases, the "classroom-and-book" model is becoming more and more inefficient.

With the advancement of the multimedia and telecommunication technologies, the implication of learning and teaching can be changed dramatically. The power of computer has enabled more information to be stored, processed, and delivered than traditional books. Meanwhile telecommunication makes the delivery of knowledge more flexible and faster than the traditional lectures in classrooms. Freed from the practical constraints that have limited teaching in the past, we can now, to a much greater extent, tailor teaching to the needs of the individual student and to the particular subject matter being taught.

In order to improve education by technology, it is worthwhile to start with looking at some commonly employed teaching methods so that we can find out where the problems are.

### **1.1.1 Traditional Methods of Teaching**

According to Roger C. Schank in [Schank, 90], six teaching methods are commonly employed by teachers today.

### **The Sponge Method**

In most educational settings, students far outnumber teachers where all that can be done is to present information and perhaps answer a question or two. This method, rooted in the practical necessity instead of any learning theory, views the students as a sort of sponge, that is, a passive absorber of information. This method encourages the students to take information without questioning. Interaction of a meaningful sort is difficult, if not impossible to be achieved in this method. When a student is not ready to or does not have the background to absorb the knowledge, this method often results in poor efficiency. However, despite its many problems, it turns out that sometimes the sponge method is exactly the right method of teaching. The trick here is to make sure that the student is ready and eager to hear what the teacher is about to tell him. When a student is curious and wants an answer, telling him the answer works fine most of the time.

### **The Apprenticeship Method**

Another popular method of teaching is the apprenticeship method. In this method, the student watches the teacher perform some task, and then tries to master the required skills under the watchful eye of the teacher. This method has been successful in teaching real-world, on-the-job skills. Unfortunately, it is far less successful in classroom teaching. One of the serious problems is that it can easily degenerate into a rote learning task where students just try to imitate without the real understanding.

### **The Artist Method**

A much less frequently used method of teaching is the artist method. In this method, the goal is to stimulate creativity as much as possible. The students are usually

given a “blank canvas” on which to print. Being different from the apprenticeship method, very little guidance is provided since the goal is to teach the student self-expression. The advantage of this method is that the students are active instead of being passive. Moreover, with the watchful eyes of a teacher who is always ready to help when the student gets stuck, even better efficiency would be achieved. However, this is extremely difficult in practice where a lot of time, patience and evaluation skills are required for from the teachers. Computer-based educational systems are promising for delivering such individualized instruction.

### **The Research Method**

In the research method of teaching, the student is asked to do research on a certain topic and present a report of his research. In many cases, a written report is augmented or replaced by an oral presentation in front of the class. To implement a research method successfully, the following requirements should be taken into consideration. First, it is important for the student to have a real interest in the topic being researched on. Second, the importance of letting the student to accrue the rewards of being an expert on the topic should not be overlooked. Third, and perhaps the most important, teachers must feel comfortable letting the students surpass their own knowledge level.

### **The Exploration Method**

One teaching method that is currently enjoying widespread use, even outside of the classroom, is the exploration method. The notion of the student as an explorer is intuitively appealing, allowing students to figure things out on their own and discovering what they need to know by finding it out for themselves. As is the case with the methods

discussed previously, the exploration method commonly falls short of the potential of which it is capable. The problem is that a teacher is required, as in the artist method, to hang around to help, encourage, and offer hints to the students when necessary for this method to reach its full effectiveness. However, too much help or too many hints can turn this method into the apprenticeship method.

### **The Argument Method**

One method of teaching that is used in only a few situations, yet still worthy of mention, is the argument method. The essence of this method is to force the student to adopt a position on some issue and then defend it. The goal is to get the student to think on his feet, to be critical and analytical, rather than to get at the truth. A study by Smith, Johnson, & Johnson [Smith, 81] found that students who study in a group where discussion and controversy were encouraged showed more accurate understanding, better retention, and higher motivation than students in concurrence-seeking groups or individual study. The most common uses of the argument method can be found in law, medical, graduate, and professional schools. The argument method is a valuable one because it teaches critical thinking. The reason that it is so rarely seen is that it is extremely difficult to implement in situations with the high student-to-teacher ratio. In another word, one-on-one based teaching, or at least low student-to-teacher ratio is a necessity to implement this method.

We have introduced the principles as well as the constraints of some commonly employed teaching methods. In all of those widely used methods, the students are passive, having weak motivations, and seldomly provided with the in-time help during

learning. This is mainly caused by the currently employed "classroom-and-book" education model with the high student-to-teacher ratio. Although some methods do have overcome the drawbacks, it is yet difficult to implement them widely in the real world.

### **1.1.2 Paradigm Shifts in Learning**

The world has changed a lot with the explosion of new technologies in the past few decades. The information technology, as a leading role in this wave of reforms, has made possible the following paradigm shifts in learning. These paradigm shifts are promising to solve the problems in the traditional teaching methods.

#### **Distance Learning**

With courses delivered through broadcasting TV systems, CD-ROMs, and distributed computer networks, learning becomes dispersed both in time and location. This provides more opportunities for those who could hardly access education resources through the traditional education system. With this change, learning becomes more accessible to all students, as well as citizens who tend to take their continuing education at home, or in their offices. Although nobody can expect that distance learning will take the place of the traditional classroom-and-book style of education, it is clear that such applications will obtain more and more position in the modern education.

#### **Multimedia for Learning**

The outburst of multimedia technologies has enabled the input, dissemination and output of any combination of text, graphics, animation, audio and video data on various media. The concept of multimedia in combining text, graphics, animation, sound and video is not

anything new. However, by integrating these audio-visual elements with the user navigating functionality, user can transform the traditional passive way of receiving information to an interactive form of controlling the flow of information to be retrieved [Grace, 95]. With multimedia technology integrated, the computer-based learning system can provide the students with a livelier and more expressive world of knowledge. However, the effectiveness of multimedia is still subject to the type of knowledge transmitted, the kind of knowledge acquirers involved, as well as the design of the transmission media presented by the author [Grace, 95].

### **Learner-centered Movement**

In the traditional education system, students are in very passive positions where they are forced to acquire knowledge page by page. Apparently, it is not very educational [Schank, 94]. Research has shown that non-linear knowledge structure is superior to the traditional linear text-based structure in terms of knowledge diffusion [Grace, 95]. The former can provide more choices in both content and method of learning so that one-on-one based teaching can be really achieved. When learners become the center of education, who can control the learning process and drive it according to their own needs and capabilities, they can learn much better. Based on the active learning theory [Schank, 94], interactive learning applications have emerged to provide a more efficient way for people to acquire knowledge [Landow, 90][Marchionini, 94].

Learning in the future is radically different from what we know now. Both of the educators and the industry are facing this new challenge. How can teaching be enhanced?

How can technology be used to create really a better place to learn? A lot of research work and experiments have been done to answer these questions. However, there are two common problems in most of the emerging computer educational software. First, the educators as courseware designers, lack computer science expertise, and try to implement long-outdated theories of learning. On the other hand, some computer science experts lack of knowledge about learning and teaching which force them to copy existing education models [Schank, 94].

In my thesis work, special attention has been paid in avoiding these two common problems. The objective is to create an innovative education environment- a really better place to learn, which integrates all the advanced information and telecommunication technologies to realize the modern learning theory.

## **1.2 Terminology Related to Multimedia TeleLearning**

In this section, some important terms that are used in the thesis are introduced to assist understanding.

- **Multimedia:** A general term implying the integration of many different media types. High speed computers, video, text, animation, audio and graphics are key components of most modern multimedia productions.
- **Hypermedia:** The ability to access mono-media and multimedia information by navigating across links [MHEG, 95].
- **TeleLearning:** Learning which allows knowledge to be stored and transferred using telecommunication technologies.

- **Distance learning:** Learning that can be done at geographically dispersed locations instead of centralized schools and classrooms.
- **Courseware:** Educational computer software.
- **Courseware authoring:** The construction of a courseware which may involves choosing the content objects, specifying the scenario, etc.
- **Linear knowledge structure:** Knowledge that is represented linearly for the learners to perceive. For example, knowledge in books is structured linearly in content, and it is assumed that learners should perceive them page by page.
- **Non-linear knowledge structure:** Knowledge that is represented in a cross-reference manner. In this structure, learners can perceive knowledge non-linearly, following the links between related nodes instead of layout pages. Non-linear knowledge structure is closer to the real world knowledge than linear structure.
- **Multimedia synchronization** [Brian, 92]: In a multimedia or hypermedia system, objects may be placed in hypothetical structures known as events. The scheduling of a certain event to take place in a specific time is called synchronization. Events are typically synchronized in one of two ways: relative to each other or relative to a schedule. Synchronization can be achieved in several levels in typical multimedia/hypermedia applications: some apply to object level structure, others to physical data streams (e.g., video). Physical data within an object may use its own synchronization method, either implied or explicit. For instance, audio and video that are embedded in the same stream may be synchronized using a proposed MPEG encoding.

## 1.3 Current Status of TeleLearning

The term of TeleLearning implies a learning environment where information can be stored and transferred distantly using telecommunication technologies. It needs the integration of technologies from various disciplines such as computer, telecommunication, information processing, etc. Before an innovative TeleLearning environment can be built, it is important to get a clear picture on the current status of TeleLearning which is to be given in this section.

According to the communication infrastructures, or say, the hardware environment, TeleLearning systems can be grouped into the following three categories. The most commonly used systems now are *broadcasting systems* which are closest to the traditional "classroom-and-book" systems. Meanwhile, the delivery of courseware through a *CD-ROM* which can be used on a Personal Computer (PC) also becomes more and more popular. Some *network-based systems* are under development and have been introduced in literatures, the most important among which are applications on the World Wide Web (WWW). The advancement of computer communication, multimedia database, satellite and wireless communication technologies may even further reshape the current situation. In this section, the advantages and disadvantages of the existing solutions are analyzed.

### **1.3.1 Broadcasting Model**

*Broadcasting model* is currently the most popular distance learning model because of its accessibility and richness in knowledge representation. Once you have a TV at home or in your office, which most of the time is connected to a cable network, you can easily attend a class by watching the broadcasting lectures. Other TeleLearning systems using video conferencing technology also fall into this category since there is always one presenter and many learners involved. However, there are also constraints for this kind of TeleLearning schema. First, learners have to follow the time schedule of the broadcasting center and cannot take the class at the most convenient time for them. Second, during the learning process, learners are always in a passive position. They cannot give any input to the presenter of the course so as to adjust the content or the speed to fit their own demands. Although some emerging TeleLearning system has tried to solve this problem by adding the telephone system for real-time communication between students and teachers, it can provide only very low level interaction. In addition, the capacity of the communication service is also limited. For example, in one of such a system called Satellite Interactive Distance Learning (SIDL) system [Citizen, 96], only three calls can be taken at a time, others will be put into a queue. This could be frustrating for a distant student trying to get a word in. On the other hand, some students may also get bored while waiting for the teacher to answer questions whose answers they already know.

### 1.3.2 CD-ROM/PC Model

With the emerging and rapid development of multimedia information processing technology, courses can be stored and delivered by Compact Disks (CDs) and finally presented on PCs. With such a model, the course material can be accessed at any time by a learner and taken in a speed most suitable for him.

Another advantage of using CD-ROM for courseware delivery is that the user can interact with the presentation through well-designed courseware. Most of the CD-ROMs have included a lot of images, sound, even audio and video for presenting the course content. Users can view the presentation, choose the one which interests him/her, and even edit them to create something totally new with their own imagination.

Compared with the broadcasting systems, CD-ROM/PC systems used for TeleLearning are more accessible, and provide higher degrees of interactivity. However, this kind of systems also has its disadvantages. First, the storage capacity of the CD-ROM is limited which results in the limitation of the quantity of knowledge to be transmitted. Second, although good one-on-one based teaching can be achieved in this schema, discussion and communication can not be provided through CD-ROM. Therefore, it is hard to build a seamless education environment. Third, as science and technology are growing fast, for some courses, it is necessary to update course content frequently to include the state-of-the-art technologies and theories. In another word, just-in-time knowledge is required to be delivered sometimes. This is especially the case in employee training courses. However, CD-ROM is read-only after being produced. Knowledge delivered by CD-ROM is static. The only way to update the content of the

CD-ROM is to throw away the old one, and order a new one. Apparently this is not very convenient and cost-effective for the users. In addition, it is a waste of time and resource.

### **1.3.3 Network-based Model**

Broadcasting model and PC model have played important roles as knowledge delivery infrastructures in TeleLearning. However, little has been done in these models to create an education environment which can provide both two-way interactions and communications between educators and learners. With the communication capability of computer network, it is possible to integrate all the advantages of the broadcasting model and the PC model to offer a seamless education environment.

The main advantage of the network-based model is the enhanced accessibility for the course as well as the seamless integration of both the information delivery and the communication capabilities in the learning environment. For example, with course material stored in database, users can freely schedule the time to take the class according to their own timetable. Meanwhile, since it is the default feature of the computer network to provide communication facilities, users can be easily linked with a professor or a colleague to discuss a topic he is interested in or get confused with. The multimedia and hypermedia information technology has further empowered the network-based TeleLearning systems. It was impossible to deliver courseware through computer network before the emergence of multimedia information processing technology. When only text can be stored and transmitted, the courseware can not successfully convey most

of the concepts to be taught. However, multimedia has provided a new dimension for the network-based TeleLearning systems to become good education environments.

A lot of TeleLearning systems are being built on the Internet- the most popular world wide computer network, for example, K-12 and WEST. They have claimed to be successful, and even started to be put into the market. A lot of experiences have been obtained from those practices. However, restricted by the network capability as well as the information coding methods, the limitations for delivering real multimedia information have not been broken through. Therefore, it cannot be regarded as the final solution for TeleLearning.

The advancement of B-ISDN and ATM technology has provided a prospective solution to deliver multimedia and hypermedia information through a computer network in a fast and quality manner. Currently, little experiment has been done on broadband network for TeleLearning applications.

## **1.4 Thesis Outline**

The major contributions of this thesis are found in Chapter 3 and Chapter 4 with the implementation details and results given in Chapter 5. The main contributions are:

- Requirement analysis and architecture design for a broadband multimedia TeleLearning system which is based on the advanced education theories and multimedia information technologies.
- Multimedia courseware authoring processes, as well as document models and object libraries for interactive multimedia courseware authoring.

- A navigator implemented on the courseware user site for school management and accessing the courseware database.

In *chapter 2*, a brief overview of the multimedia and hypermedia information system is given. Two applications are described as examples to show the potential of hypermedia and multimedia systems. Also, the international standards on multimedia and hypermedia information coding and delivery are introduced. *Chapter 3* introduces the design of a broadband multimedia TeleLearning system. The design rationale is illustrated first, followed by the description of the system architecture, and then the courseware processing models and strategies. Courseware authoring is another aspect which is important to ensure efficient courseware services to be offered. *Chapter 4* presents the issues on courseware authoring. First, the general courseware production process is introduced. Then models and details of courseware authoring methodologies in our system are provided. Two document models are designed for the courseware authoring. Moreover, a basic MHEG [MHEG, 95] object library and a multimedia courseware object library are designed for the convenience of the object coding. We have also implemented a courseware navigator as part of a distributed multimedia TeleLearning system, and this is introduced in *chapter 5*. Some details of implementation as well as a sample learning session is given. Finally, in *chapter 6* we conclude and suggest directions for the future research.

## Chapter 2

# Multimedia and Hypermedia- Applications and Standards

In chapter 1, the TeleLearning systems are categorized by their hardware environment. However, for the computer-based TeleLearning system, information-processing model is another important issue. In a computer system, knowledge can be represented by various data types such as text, image, audio, video, etc. To represent a complicated idea, this coded information should also be organized logically or semantically to form a document. The representation of knowledge in a TeleLearning system is called information coding, while the organization of the coded information is the issue of document processing. Information processing is one of the key technical aspects of the computer-based TeleLearning applications.

## **2.1 Multimedia and Hypermedia Applications**

Multimedia and hypermedia has broad range of applications. Simulation and games can be made much more realistic by the use of multiple media in computer based systems. Examples are flight and maritime simulators. The computer processing of audiovisual information can make it possible to extend the audience for theaters, museums and other culture events. Much material that is currently published on paper could be enhanced by the integration of audiovisuals and by being available possibly in a hypertext manner over a telecommunication network, e.g., tourist guides and yellow pages. More applications can be developed in sales and advertising, office information systems, medical applications, interactive television applications, etc. In a word, the application domain of multimedia and hypermedia information system is almost limitless. The following two are examples to further illustrate the potential of multimedia and hypermedia applications.

### **2.1.1 The Hyper-News at 11:00 Application**

In [Brian, 92], Brian has described a Hyper-News application as follows:

"Jon is working late on a computer program in his office. In front of him, several windows are open on his workstation; some contain small pieces of C language source code, others output from his recalcitrant executable. The time display in one window is updated to read 23:00, and an alarm bell rings. Jon knows that this means that it is time to receive the news of the day.

After clicking on an Icon, Jon is given a menu of news stories. A piece about strife in his native Lithuania catches his eye, and he uses the mouse to select it. The face of an incongruously ebullient newscaster appears in one window, in another, a scene of police clashing with demonstrators. The workstation's built-in speakers vibrate to the chirpy tones of the newscaster's voice. At the bottom of the display, the otherwise solemn text of the story scrolls by. Buttons allow Jon to display the text of related newspaper and magazine articles.

Jon decides he's better off finishing his program and, with an unceremonious click of his mouse button, relegates the news application to an icon."

### **2.1.2 A Digital Library Application**

Another example to demonstrate multimedia/hypermedia applications can be the digital library application which is one of the hot spots among all hypermedia applications.

A student sits in front of his PC, calling via modem, to a network to a library and trying to find some information. On that network being searched through, a variety of CD-ROM disks holding information are running in CD-ROM drives. On the screen, the student can choose by category, keyword, author, or even media. Usually the interface simulates the virtual reality of a real library. Also, the student can find out many related documents by just clicking a term or phrase which interests him/her. On finding the information required, the student can pull it out from the storage, download it, and then, log out the network [Greg, 93].

We all have the experience of wandering through piles of shelves, climbing up and down in a real library, while sometimes still can not find what we want. However, in a digital library, students can do all this work without even leaving their desks. On the other hand, all the information stored digitally can be shared by a big amount of users at a specific time provided that the network has enough capacity. There is no borrow, return or hold kind of work which usually causes a lot of work and inconvenience to both users and librarians.

## **2.2 Standards and Methodologies for Multimedia and Hypermedia Information Delivery**

Any process can be virtually represented using hypermedia techniques, and therefore the real number of applications is almost limitless. It is deemed necessary to make interchange formats open and extensible. Moreover, as an information-processing model mostly suitable for information system of large scale which may include platforms as well as software from different vendors, the information interchange format of hypermedia should be standardized. Several international standards have addressed the information coding and document processing for multimedia and hypermedia. The following are brief introductions on the most important two among them- HyTime and MHEG standards.

## **2.2.1 HyTime: Hypermedia/Time-based Structuring Language**

HyTime is an international standard which was developed principally by ANSI committee X3V1.8M, and subsequently adopted by ISO in April 1992. It defines a model and language for the representation of “hyperdocuments” that link and synchronize static and time-based information contained in multiple conventional and multimedia documents and information objects. The language is known as the “Hypermedia/Time-based Structuring Language”, or “HyTime”. The field of application of HyTime is Integrated Open Hypermedia (IOH)- the “bibliographic model” of hyperlinking wherein an author can, by a suitable reference, link to anything, anywhere, at any time [HyTime, 91]. It is intended for use as the infrastructure of platform-independent information interchange for hypermedia, as well as synchronization and non-synchronization multimedia applications. Application developers will use HyTime constructs to design their information structures and objects, and the HyTime language to represent them for interchange.

Some important principles and concepts of HyTime standard will be described in the following context of this section.

### **2.2.1.1 HyTime and SGML [John, 93]**

SGML standard (ISO 8879-1986) is a document architecture for text documents. It defines a language for describing the logical structure of a text document using standard markup. This markup specifies what parts of the document belong to what document components, or SGML elements. Each SGML document uses a Document

Type Definition (DTD) which specifies element types, attributes as well as the hierarchical relationship among instances of element types in a document. Typically a DTD defines a whole class of documents, and many document instances share a common DTD.

HyTime standard is an extension to SGML so that markup and DTDs can be used to describe the structure of multimedia documents. It defines a set of architectural forms for the definition of multimedia DTDs. Each architectural form specifies how an SGML element type can be defined in a DTD whose instances contain information regarding certain HyTime concepts or groups of concepts.

In the spirit of SGML whose markup describes logical relationships but not presentation attributes, HyTime does not provide a representation for aspects of a hyperdocument which are specific to a particular media type or user interface. HyTime with SGML provides a means for defining the representation of hyperdocuments as character files which can be interchanged between and processed on any platform. It also provides a layered model for hypermedia document processing.

HyTime can be viewed as a meta-language for defining an arbitrary number of multimedia document types. Apart from SGML syntax, it implies a data model for the space of structured time-based multimedia documents.

### **2.2.1.2 HyTime Modules [John, 93]**

HyTime is designed to be used modularly. There is one required module and a number of interdependent optional modules (Figure 2.1). A HyTime application can incorporate just the modules which contain the architectural forms it needs and omits the

rest. Every HyTime document state what modules and options are needed for its processing.

The *base module* is required by all of the other modules. It includes facilities that underlie the facilities of other modules or that are (sometimes optionally) available regardless of which of the other modules are supported. These facilities include:

- *Hyperdocument management facilities*: required for all other HyTime facilities.
- *HyTime identification facilities*: permit the replacement of HyTime specific identifiers with user-defined identifiers.

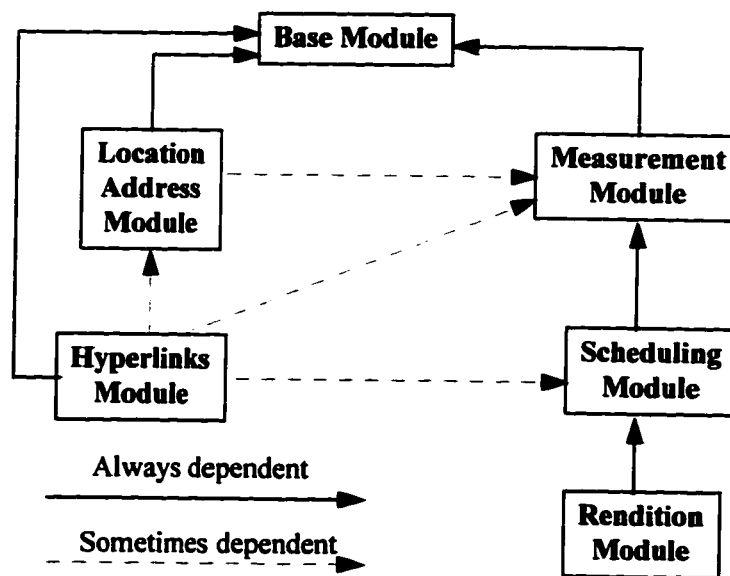


Figure 2.1 HyTime module interdependencies [John, 93]

- *Means for specifying application-defined expressions, called xenofoms*: a required facility. Xenofoms are combined with other HyTime architectural forms as required by the application.

- *Coordinate addressing facility*: allows the dimensions and positions of events, etc., to be scheduled, and allows document locations to be addressed by position. This facility is required to support location address module and/or scheduling module.
- *Optional means for specifying activity-tracking policies*: Any element in a SGML document can have an “active-tracking” attribute, which functions as a pointer to an “active-tracking policy” element. The active-tracking policy element can associate policies with any combination of the following activities: creating, modifying, linking, accessing, and deleting.
- *Other optional basic utilities*: intended to provide syntactically economical means of declaring default attribute values and definition tables.

The *measurement module* gives a document the ability to represent concepts involving dimensions, measurement, and counting. Dimensions can be defined which use particular units of measure, and document objects can be placed along these dimensions at particular locations which are specified by these measurement units.

The *location address module* provides various means of specifying locations in a hyperdocument which could not be specified by SGML alone. The details of HyTime addressing can be found in section 2.2.1.3.

The *hyperlinks module* supports the defining of links between portions of document which can be traversed by a user in hypertext fashion. The specification of link endpoints can involve the location address, measurement, and scheduling modules.

The *scheduling module* places document objects in *Finite Coordinate Spaces (FCS)*, which are defined as collections of axes. Because of the use of dimensions and other related concepts, this module is dependent on the measurement module. Events are located on the axes of a FCS. The boundaries of these events can be specific numbers along these axes or can be related to the boundaries of other events within the same FCS.

The *rendition module* requires and expands upon the scheduling module. It specifies how events in one FCS can be mapped to another FCS. Typically the first FCS provides a generic representation for a collection of document objects while the second FCS specifies the layout for a particular presentation of those objects.

### **2.2.1.3 HyTime Addressing**

Addressing is a basic function provided by HyTime which includes two parts: the hypermedia part and the time-based part.

#### **The Hypermedia Part**

The essential functional requirement of an Integrated Open Hypermedia (IOH) system is the ability to create a link to any arbitrary portion of any kind of information object at any time. HyTime, providing combinatorial facilities to build up complex chains and aggregates of locations that can be linked to as a single address, meets this requirement. The HyTime forms of address are (Figure 2.2):

(1) By a unique name in a name space, for example:

- A universally unique public identifier: Champion Rutherford's Pride of Witheringham (Registered AKC) [HyTime, 91].

- An equivalent locally unique name: Spot [HyTime, 91]

This is the basis of hyperlinking. It is the most robust form of address in that it can survive changes in the object being addressed.

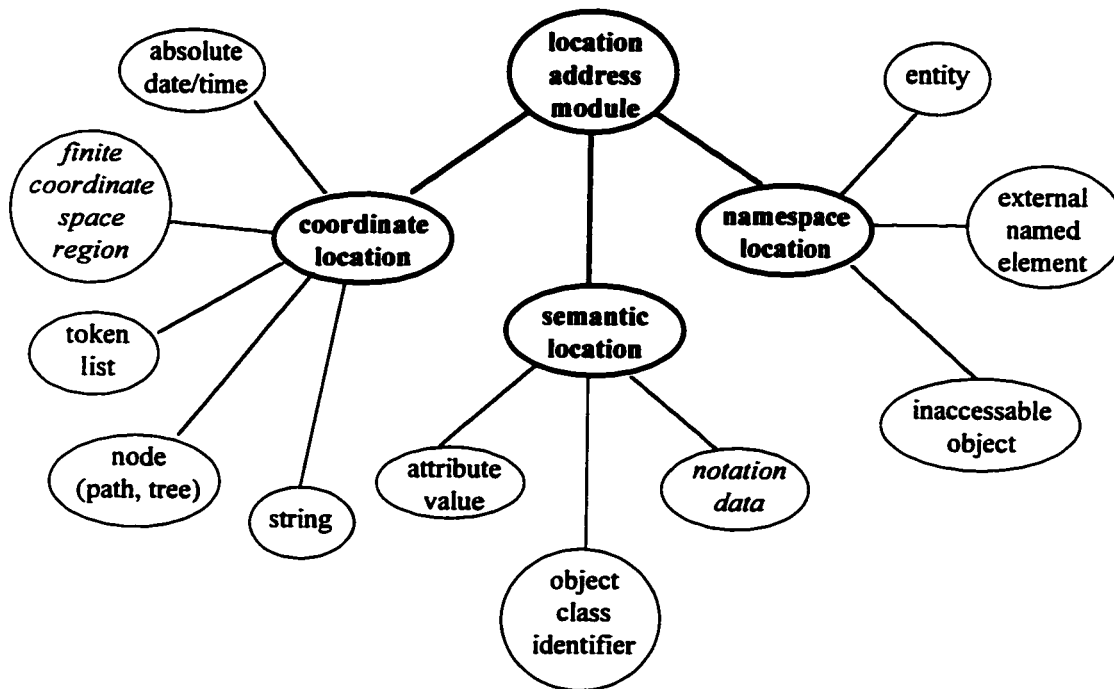


Figure 2.2 HyTime location address module [Steven, 91]

(2) By a position on an axis in a coordinate system, for example:

- The second house from the end of the street.
- Next Monday at this time.
- The first child of the second child of the root of the tree structure (and, by analogy, “his first grandchild by his second child”).
- Five minutes ago.

This is essential to scheduling, projection, alignment, and synchronization. It can be used for unnamed objects and arbitrary portions of objects.

(3) By a semantic construct, for example:

- The book I read yesterday.
- The boy in a red hat.
- The world's most beautiful garden.

This kind of addressing is necessary in order to provide addressability for arbitrary data. HyTime passes semantic addresses to interpretation programs to convert to a name space or coordinate address.

HyTime provides a complete schema for all the above three forms of addressing, and the ability to convert coordinate and semantic addresses to name space addresses so that all the three can be linked uniformly. In addition, it also supports the creation of higher forms of addressing by applications. Access from HyTime address of a location to its physical storage is provided by the SGML entity manager component of the system, which might invoke network processes to retrieve the located data from a remote database.

### **The Time-based Part**

HyTime's coordinate addressing is the foundation for event scheduling, alignment, and synchronization, all of which involve the manipulation of coordinate addresses. Synchronization can be represented, for example, by establishing a temporal unit (such as second) as the system of measurement for a coordinate axis, and specifying the address of one event as a function of the coordinate address of another event.

HyTime does not address the representation of audio or video content data, but simply defines the means by which the start-time and duration of such data can be synchronized with other digitized information. Nor does it specify the layout process by

which occurrences of unformatted documents and other information objects can be made to fit the positions and extents specified for them.

#### 2.2.1.4 HyTime Document Processing Model [Steven, 91]

The conventional HyTime processing model puts the application in control of everything that happens. When it is time to process the HyTime document, the application calls the HyTime engine, which in turn calls the SGML parser. As it is parsing the document, the parser informs the HyTime engine about everything that it encounters (Figure 2.3). While this is going on, the HyTime engine does two things:

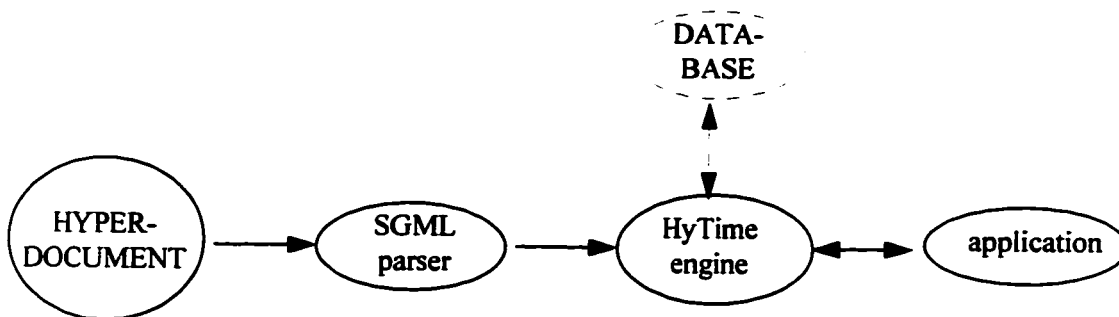


Figure 2.3 HyTime document processing model [Steven, 91]

The engine passes the entire output of the document back to the application. The application uses this information as it sees fit. However, it may also ignore it entirely.

The engine creates an engine-internal data structure based on the HyTime-specific information arriving from the parser.

After the document has been parsed, the application may query the HyTime engine in various ways. The engine assumes responsibility for determining where things

are on FCS schedules, for resolving document location elements to the data they indicate, etc.

## **2.2.2 MHEG: Multimedia and Hypermedia Information Coding Expert Group**

MHEG stands for Multimedia and Hypermedia Information Coding Expert Group, which is also known as ISO/IEC JTC1/SC29/WG12. The standard it provides is Coded Representation of Multimedia and Hypermedia Information, commonly called MHEG standard (ISO/IEC DIS 13522-1). MHEG is a developing International Standard (IS) which provides a coded representation for multimedia/hypermedia information to be used and interchanged in real-time by applications in a wide range of domains and on heterogeneous platforms. Examples for the applications are interactive multimedia applications, messaging systems, document interchange services, etc. The interchange can make use of storage, local area network, wide area telecommunication network, or broadcasting network. MHEG standard aims at defining a common base for many of the multimedia and hypermedia applications which will be developed in the forthcoming years in different fields. This common base is the coded representation of independent and elementary units of information, which will be specified as objects, and handled or interchanged by the different applications making use of them. The standard is aiming at supporting real time interchange and presentation using minimal resources. It has designed a simple and powerful encoding format while does not enforce any semantic

interpretation by the using application. MHEG focuses on inter-media synchronization. The output format for interchange is ASN.1 or SGML.

The object-oriented approach was chosen for the design of the standard because it fits the requirements of active, autonomous and reusable objects. However, the implementation of the MHEG standard is not enforced to be based on an object-oriented system. Messages for the communication between MHEG objects, or for the communication of these objects with client applications, are specified in the standard, but it remains up to the applications or services to define the way to use them or handle and encode them if necessary.

The work on MHEG began in ISO in 1989 as an ad-hoc group led by Mr. Francis Kretz of France Telecom, following an initiative by Dr. Hiroshi Yasuda of NTT. The committee is now ISO working group JTC1/SC29/WG12, also known as "MHEG". This group is part of the well known "JPEG, JBIG, MPEG" activity of SC29. The standard is developed in a number of parts. Part I (ISO/IEC 13522-1) provides an object oriented approach and is now the basis for the work on interactive digital TV. This part has already reached Draft International Standard (DIS) stage. A proposal for Part II of the standard has been submitted recently to the working group which uses HTML as an alternative notation for information interchange. Part III is to provide MHEG extensions for scripting language support. Part IV covers the registration procedure for MHEG format identifier. Both part III and part IV is still in initial stage. Part V of the standard, developed to support the distribution of interactive multimedia applications in a client/server architecture across platforms of different types and brands, reached its DIS stage in December 1995, and is scheduled for promotion to IS in July 1996.

### 2.2.2.1 MHEG Information Coding- Classes and Objects

MHEG standard uses object-oriented methods to achieve reusable and autonomous information structures. It defines a classification of structures based on an analysis of the common behavior and properties of multimedia/hypermedia information. Inheritance and overridden is supported in subclasses. Each object has associated with it some methods, or procedures which control the presentation of the object. The attributes associated with the object form the parameters of the methods. The methods would normally be associated with a class of object, such as digital audio. MHEG makes no attempt to define the nature of the method: it is a system and application design issue. Methods may be entire programs, system services or routines in an application.

There are eight object classes defined by the standard. Nevertheless, application designers can add new classes into the system. From these classes, MHEG objects may be instantiated by the object designer and interchanged between applications. The defined eight classes can be divided into the following categories.

#### **Information Coding**

There are four classes which code information into objects. These are also called *model classes*. Run-time objects which are copies of model objects and exist only in the presentation system can be created for presentation by the object designers. The *Content Class and Multiplexed Content Class* contain or refer to the media objects with a parameter set specifying characteristics for content presentation. This parameter set may contain an identification of the coding method and may specify the original size, duration and volume of the data. The values are expressed using generic units. *Multiplexed*

*Content Class* is a sub-class of the content class which contains or refers to the multiplexed media objects with a description of each multiplexed stream. The *Composite Class* provides facilities for associating multimedia and hypermedia objects with a consistent approach of synchronization in time and space, or linking of a set of objects. *Composite MHEG objects* may also contain other composite objects. The *Script Class* defines a container for specifying complex relationships between MHEG objects and run-time objects by a non-MHEG language.

### **Synchronization and Interaction**

There are two kinds of relations among objects. One is the synchronization in time and space for presentation. The other is user interaction where a click can lead to the status change of another object. Both of them are achieved by link class and action class. The *Link Class* defines a structure which specifies a set of relationships. Each relationship is defined between one or more “sources” and one or more “targets”. The actions, which are described by the action objects, are to be applied on certain targets when the conditions are satisfied. Instances of the link class are used to specify the time sequencing, spatial positioning or logical interaction between MHEG objects or run-time objects. The *Action Class* objects can be used alone or within a link object to describe the link effect. It defines a structure which specifies a synchronization set of elementary actions to be applied on one or more targets. The elementary actions defined include “Preparation”, “Creation”, “Presentation”, “Rendition”, “Interaction”, and “Activation”.

### **Interchange**

There are two classes defined for the purpose of interchange. The *Container Class* provides a container to regrouping multimedia and hypermedia data in order to interchange them as a whole set. The *Descriptor Class* defines a structure for the interchange of resource information about a single or a set of other interchanged objects. The information can be used to facilitate a correspondence between the resources required to present the objects and the resources available to the system, or to perform a negotiation between the source of the MHEG objects and the presentation site.

#### **2.2.2.2 MHEG Object Processing Model [MHEG, 95]**

MHEG objects exist in different forms while being processed. This is a key basis for designing an MHEG system. Figure 2.4 illustrates the MHEG objects life cycle.

##### **Form (a) objects**

This form is the interchangeable multimedia/hypermedia information, encoded in ASN.1 or SGML. It is generated from the MHEG object encoding. Objects are created by a “CODER” in the MHEG engine. While an MHEG object of form (a) is received by an MHEG engine at the presentation site, it is converted into form (b).

##### **Form (b) objects**

The MHEG objects are decoded into form (b) after they are received by the presentation site. Form (b) is the internal format for MHEG engine processing. This form of objects remains in existence until a ‘destroy’ action is applied to them.

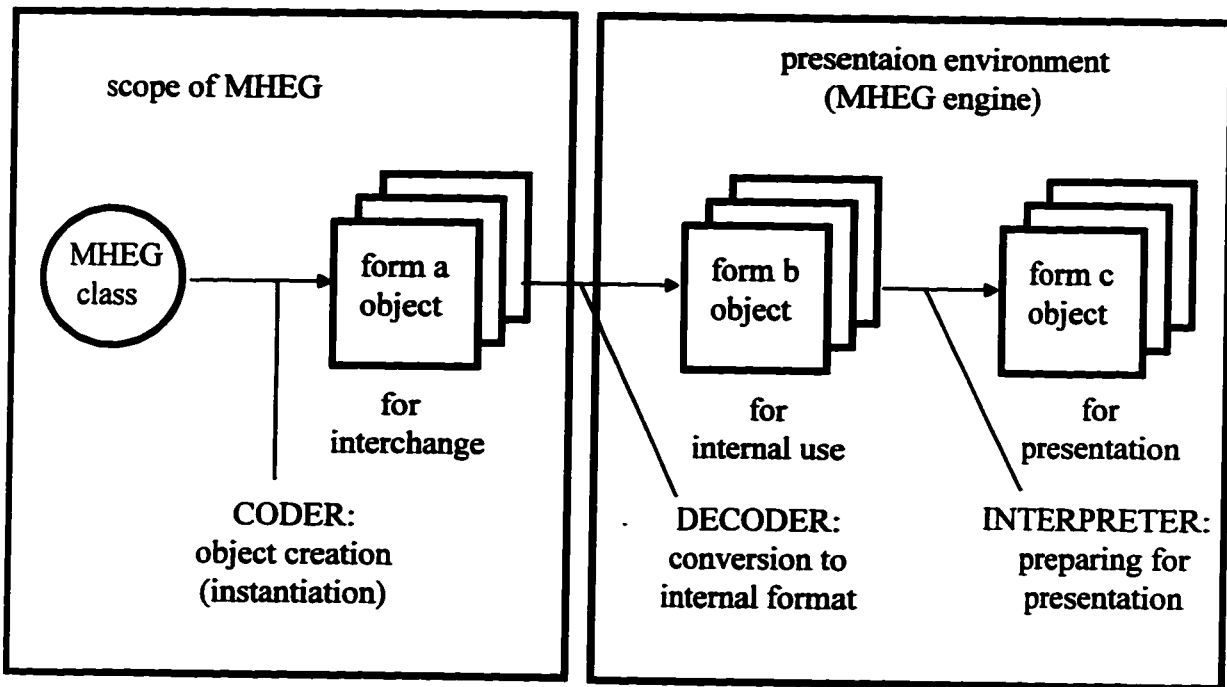


Figure 2.4 MHEG objects life cycle

### Form (c) objects

Objects existing in this form are run-time objects. They are copies of form (b) objects, used for presentation in the end-user environment. Form (c) objects come into existence whenever a 'new' action is applied to an appropriate form (b) object (i.e., a model object). The result is a copy of this object, but can be presented and may have attribute values changed. Form (c) objects are removed from existence by a 'delete' action.

MHEG defines run-time object for the purpose of reusing model objects in different presentations. A run-time object which corresponds to a specific view of the 'model' data or composition, is different from the interchanged model object containing the reusable data or composition. It is the object designer's responsibility to create the

runtime-objects using the MHEG action 'new'. The presentation or activation of a runtime-object does not affect the model object, which allows the reuse of a same model object in different runtime-objects.

There are four kinds of runtime-objects defined according to the model objects from which they are generated. They are runtime-script, runtime-content, runtime-multiplexed content, and runtime-composite. A runtime-content, a runtime-complex content or a runtime-composite is also called a runtime-component. A socket is an element of a runtime-composite where a runtime-component is plugged into. Different types of sockets are defined according to the runtime-component which is plugged into the socket. These include:

- Empty socket, i.e., a null runtime-component is plugged.
- Presentable socket, i.e., a runtime-content or a runtime-multiplexed content is plugged.
- Structural socket, i.e., a runtime-composite is plugged.

MHEG standard has defined object-processing method by the object life cycle introduced above. Since the MHEG systems are developed for the processing of the information coded in those objects, it is apparent that the object-processing method should be the basic feature of an MHEG system. It should be noted that form (b) and form (c) objects live within a presentation environment (e.g. the MHEG engine). This implies that they are assumed to be extinct whenever the presentation environment vanishes.

### 2.2.2.3 MHEG Synchronization

MHEG identifies four levels of synchronization for multimedia objects:

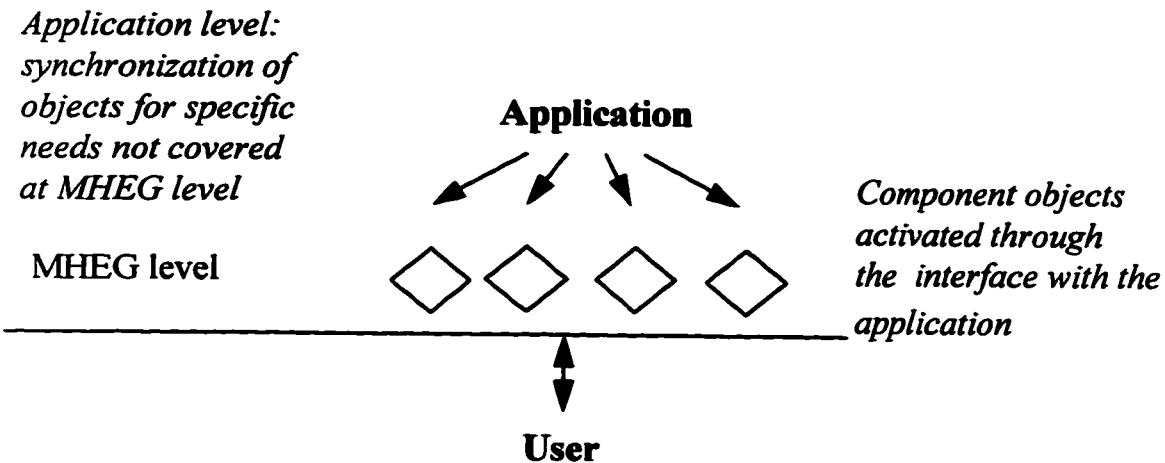


Figure 2.5 An example of synchronization defined at application level

1. Script defined by a using application: See Figure 2.5. The script may contain complex synchronization taking into account previous user replies, calculated values, and the state of system resources, e.g., the overall view of how a course is to be taught, or a product to be presented. Script synchronization is to be provided by future international standards such as AVI scriptware.

2. Conditional synchronization at MHEG level: The current state of the objects in the presentation may trigger a reflex action on another object, e.g., “When the audio has finished, display the image”.

3. Spatial-temporal synchronization at MHEG level: This specifies the position in time and space of one object with respect to another, e.g., “show the product name 2cm above the image”.

4. Inter-media synchronization provided within the object: In this case, the data that compose the object are already glued together in a non-redefinable way. It is

synchronization at the system level, e.g., synchronization of the audio and video streams within an MPEG audiovisual sequence. This case is out of the scope of MHEG.

### Spatial-temporal Synchronization

For the requirements of temporal synchronization, the MHEG standard defines a general mechanism which describes the inter-objects relations based on distinction between serial and parallel relation. There are several kinds of mechanisms:

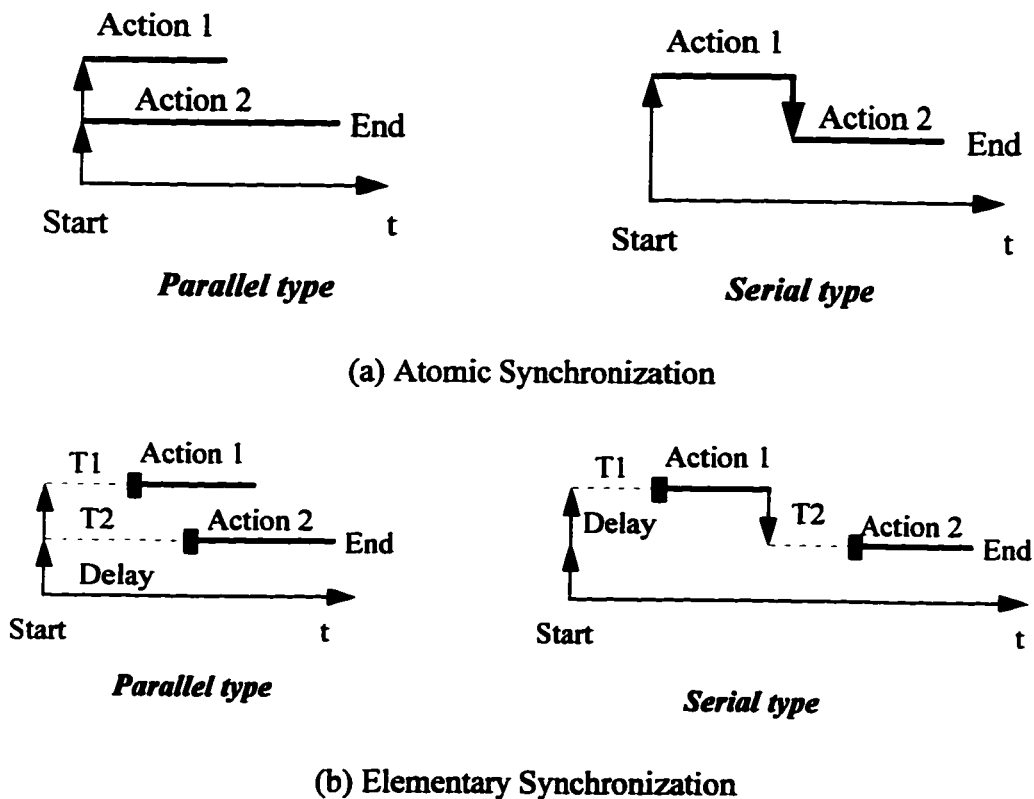


Figure 2.6 An example of atomic synchronization and elementary synchronization

1. Atomic synchronization: Only two component objects of a composite object are involved in. These are particularly simple modes of relations. See Fig. 2.6a.
2. Elementary synchronization: correspond to a more general description of the relation of two objects, to be presented simultaneously or sequentially. Two component

objects (Action 1 and Action 2) and two time values (T1 and T2) which are associated with the objects are specified. See Fig. 2.6b.

3. **Cyclic synchronization:** correspond to the repetitive presentation of an object, which allows events to be synchronized to some periodic events, such as clock tick.

4. **Chained synchronization:** allows basic objects to be chained together into a new composite object for presentation and synchronization with other streams [Brian, 92].

### **Conditional Synchronization**

The MHEG standard provides a general mechanism for the representation of specific input objects, by combining conditions and describing the resulting actions. The concept of conditional actions set is thought to be suitable for such a need. There are two types of condition:

- *Trigger conditions:* The trigger is activated when the MHEG engine detects a change in the value of an object status or a presentable status.
- *Additional conditions:* The MHEG engine is required to test the value of one or more additional status.

An example for the hyperlinking would be an application which specifies that an audio segment is played when some push-button is activated. A hyperobject would be declared which has as its component objects the audio segment and the push-button. The attribute for the push-button would be defined such that activation of the audio object is predicated upon the button's activation.

### 2.2.2.4 Multimedia/hypermedia Interchange

#### MHEG Interchange Model

As we know, MHEG standard is intended to provide interchange facilities for various media types which can be used in heterogeneous platforms. However, a using application may have various levels of complexity of data interchange. It is important to know the interchange model which is shown in Figure 2.7.

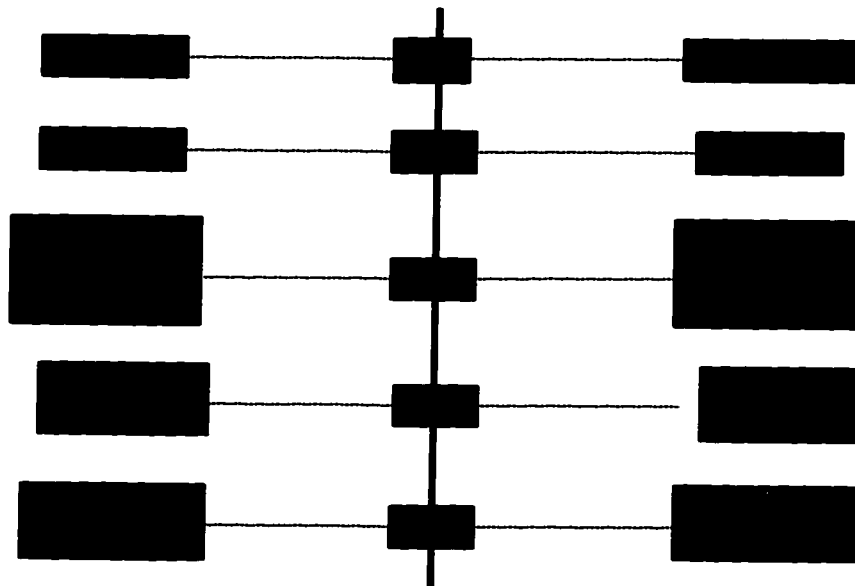


Figure 2.7 Multimedia/Hypermedia interchange model

The data exchange at the application level (A) is not covered by this standard. The exchange at the Script level (S) is covered by standards for hyperdocument and scriptware interchange. It is the subject of MHEG standard to define data exchange behaviors at the MHEG Object level (M). The Non-MHEG Content data level (C) is addressed by those individual mono-media recommendations and standards such as MPEG, JPEG, etc. The Other Protocol Element level (OPE) specifies the exchange of

elements such as messages and acknowledgements required by the application. Neither is this level addressed by this standard.

### **Container and Presentation**

In this standard, the MHEG objects are interchanged within containers. The container concept enables the grouping of a set of multimedia and hypermedia objects interchanged between systems and applications (Figure 2.8).

When an object is to be used in a presentation, a further preparation action is required before the object can participate in a presentation. The concept of *container* and *presentation* are represented within two generic tools, the container class and the composite class. The container class is used as an information-packing tool to convey multimedia and hypermedia information, while the composite class is used as an information presentation tool under the control of a multimedia and hypermedia scenario.

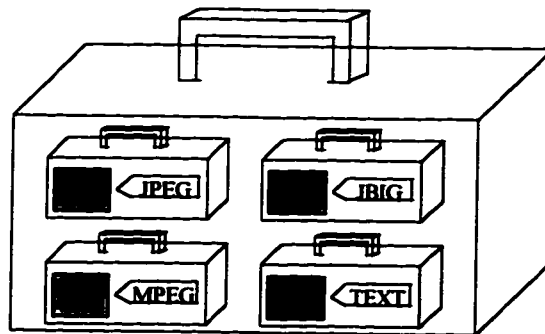


Figure 2.8 MHEG for interchange and identification of multimedia information

### **MHEG Engine for Interchange**

The more specific model for interchanging MHEG objects between two systems- 'A' and 'B' is shown in Figure 2.9.

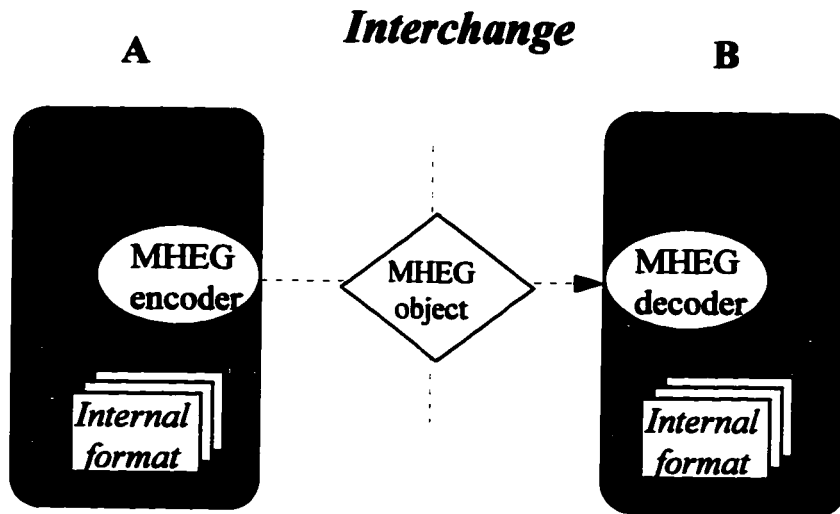


Figure 2.9 Interchange of MHEG objects [MHEG, 95]

In the above figure, MHEG object is only coded at the interchange point between the using applications (A and B). The MHEG encoder converts the internal format used in A to the MHEG format, while the MHEG decoder decodes the MHEG object to its own internal format after the object is received.

### 2.3 Comparison of HyTime and MHEG

HyTime and MHEG are two important International Standards for multimedia and hypermedia information delivery.

At first glance, the two standards may appear to be conflicting since both define formats for the representation of hypermedia information and are intend to facilitate the interchange of such information between applications. However, these two standards had different scope and purposes while being developed which results in their different perspectives as well as methodologies in addressing information-processing issues. HyTime is a standard for storing and interchanging multimedia documents. It is designed

specifically for the publishing applications, while MHEG is aimed at real-time, interactive and final form information interchange and presentation in a wide range of applications. On the other hand, in some specific applications, MHEG and HyTime may even assume complementary roles.

A lot research is being done on the differences as well as the relationships between the two standards. Moreover, there are ongoing efforts on developing the standard interface to join them to be used in one application. A potential approach is to use MHEG as the output format for hypermedia application taking HyTime as input. This would benefit from both the expressive power of HyTime and the runtime efficiency of MHEG. [MultiTorg, 95]

In this section, I am going to compare the two standards under the consideration of different application domains.

### **2.3.1 Multimedia and Hypermedia Authoring and Publishing**

HyTime is more suitable than MHEG for multimedia and hypermedia document authoring applications. As the extension of SGML, which is the current publishing standard, HyTime has defined a document logical structure as well as a language for hypermedia document processing. However, MHEG is by its definition designed for final form interchange. In MHEG, multimedia and hypermedia objects are coded and represented with the purpose of direct interchange and presentation. Additional processing of their structure is not required. Consequently, if an application (e.g., an editing application) needs to modify the objects from the MHEG interchanged final form,

this application will have to apply specific methods for the updating of the objects' components, or possibly to translate the interchanged form into its own internal form for editing purpose. Therefore, MHEG is not as suitable for hypermedia authoring as HyTime.

### **2.3.2 Real-time Interchange and Presentation**

HyTime may not be suitable for real time interchange and presentation of hypermedia document, whereas MHEG is by definition. The MHEG standard has no control over and makes no assumptions about the quality or capability of the underlying networks through which applications may interchange MHEG objects [MHEG, 95]. It deals with these uncertain circumstances in two ways:

It allows the author to define a set of specifications that would necessarily have to be satisfied to ensure real time applications regardless of the interchange media capabilities.

It defines an MHEG descriptor object which provides the following: a description of objects to be supported by the system, a description of media encoding, a 'system readable material' and a 'readme' mechanism by which communicating applications can negotiate an optimum interchange session.

Meanwhile, MHEG objects code information in final form for direct presentation which further ensures real-time presentation. For example, links in MHEG link objects are fully resolved and require no further processing other than their direct execution.

### **2.3.3 Interactive Applications**

MHEG is more suitable for interactive applications than HyTime. The MHEG objects which are used to set up complex documents or programs, must encapsulate not only synchronization between mono-media information, but also interactivity between input objects and other objects through basic linkage mechanisms (e.g., “go to link”). One important feature of MHEG is that it provides facilities to describe results of user interactions rather than provide interaction facilities. All interaction behaviors are based on generic selection and modification behaviors of run-time components and sockets. In HyTime, there is no explicitly defined mechanism to provide interaction between end user and the application during presentation except for hyperlink. This is much more limited comparing with the rich interaction styles that MHEG can provide.

## Chapter 3

# **A Broadband Multimedia TeleLearning System**

In this chapter, we first analyze the requirements of a broadband multimedia TeleLearning system based on modern education theories. Then technical solutions are chosen for the system design. After that, the system architectures, details of each component, as well as the courseware processing strategies in the system are described.

## **3.1 Design Rationale of a Broadband Multimedia TeleLearning System**

We have surveyed the current status of both the infrastructures and the multimedia/hypermedia information delivery standards for a TeleLearning system. However, blindly exploiting those new technologies does not mean it can result in a good place to learn. As described in the previous chapters, each technology has its own advantages and constraints. What kind of system can be regarded as an innovative and

truly better TeleLearning system? What technical solutions should be chosen for such a system? These are questions to be answered before the system design.

### **3.1.1 Requirement of an Innovative Multimedia TeleLearning System**

We conclude the requirements for an innovative multimedia TeleLearning system to be as follows:

#### **Course On Demand**

In traditional education systems, the students take classes in a classroom on a specific time set up by the school administration system. It is a very common case that the students encounter with time confliction to attend the class, especially for those who are part-time students. Moreover, in some special cases, for example, bad weather or vehicle malfunctioning, they have to miss a class without any other choices. Even when they are able to attend the class, the course material or the presentation can be totally unfit for their learning style or cognition ability which may result in bad efficiency in learning. The education and training systems are not satisfactory enough in accessibility and flexibility.

Provided with the rapid advancement of the telecommunication technology, it is possible now to have a TeleLearning system that can really achieve one-on-one based education. We propose a Course On Demand (COD) system for eliminating the drawbacks in the traditional education systems. In a COD system, courseware is stored in a database after being created, and is provided on demand for the presentation on an end-

user system. With such an infrastructure, the students now have the flexibility to access the knowledge repositories to take a class from any access point in a network at a time convenient for them. A terminal at home or in the office can provide an integrated learning environment which may take the place of the traditional classroom. Moreover, through the interaction mechanism provided in the courseware presentation, the students can guide the knowledge diffusion process and tailor the content and the learning speed to their own learning style and cognitive ability. In addition to the above, help may be available on demand to facilitate the learning process. When a student encounters a problem during learning, he can always get facilitation on demand in a fast and satisfactory manner.

#### **Learner-Centered Education Environment**

In the traditional education system, the teacher controls the whole learning process which includes organizing lectures, assignment, as well as exams for a class of students. Teachers are the center of the learning process. However, based on active learning theory [Schank, 94], students should be provided the flexibility to control the learning process according to their own demand. In another word, the student should be the center of the knowledge acquisition process so as to learn most efficiently. In such a learner-centered education environment, they are motivated by a clear goal, learn by practicing or browsing intentionally, and finally find out answers or get the skills they are seeking to obtain. Consequently, the role of a teacher changes to the learning facilitator whose work may include courseware production or on-line consultation.

### **Integrated, Open and Scaleable Environment**

Courseware involving different topics, designed for different student profiles with different learning objectives requires the use of different learning schema. For example, to teach ATM technology to fourth year university students or trainees of a telecommunication company can be of many differences. Consequently it may result in choosing of the different teaching schema in the courseware. A good TeleLearning system should be general enough to include not only various teaching architectures [Schank, 94], but also various learning paradigms such as lectures, libraries, discussions, exercises, etc.. The system should be developed modularly and able to be constructed according to the demands of different courses or students. Communications between the students and the professors should be achieved by means of real-time multimedia conferencing system, e-mail, or telephone.

### **Courseware Reusability**

Multimedia courseware system is a large and complex system. For many years courseware authors have tried to determine the factors which influence the efficiency of courseware development. Reusable, instructional templates within authoring tools were identified as contributing to efficient courseware authoring over fifteen years ago [Avner, 79]. Courseware reusability can be also achieved in storage and delivery process of a courseware.

In summary, an innovative multimedia TeleLearning system should provide accessibility in different time and location, flexibility for individual learning requirements, interactivity for learner-centered education, modularity for different scale

of system, and reusability for efficient courseware processing. These are also the design objectives, as well as the characteristics of the system designed in this thesis.

### **3.1.2 Choosing Technologies**

In the last section, we have analyzed some requirements for an innovative and advanced multimedia TeleLearning system. In the Multimedia Information Research Laboratory (MIRLab) at the University of Ottawa, we are developing a Multimedia Interactive TeleLearning System (MITS) on a broadband network which can achieve the above described requirements. In designing the system, the following issues have been considered.

#### **3.1.2.1 Communication Infrastructure: Broadcasting, CD-ROM/PC, or Computer Network?**

As introduced in 1.2, there are several communication infrastructures commonly used by TeleLearning applications. Among all those infrastructures, computer network is the most suitable to meet the requirements we have defined.

When a TeleLearning system is build around learners instead of educators, the broadcasting model cannot provide any longer services as required. The problem of this model is that the playback of the course cannot be driven by each user on a one-on-one basis. Therefore high level interactivity can not be achieved. Apparently, a course presentation on a stand-alone computer can provide an environment where learners could be made the center of the learning process. However, it is still not good enough to meet

the requirement specified because of its limitation in storage capacity, the lack of communication and discussion facilities, as well as the difficulties in course content updating and school management in such systems.

Network-based model has advantages in providing large information repository, interactive course presentation, seamless learning environment which may include all the traditional as well as modern learning paradigms, high degree of reusability in course processing, and easiness for school administration. Therefore, it provides an infrastructure to build applications on which all the services and characteristics described in 3.1.1 can be achieved.

### **3.1.2.2 Information Processing Techniques: HyTime or MHEG?**

A network-based TeleLearning system is a complicated system which may include many network applications such as courseware authoring/rendering, courseware database management, multimedia conferencing, etc. The information in the courseware is coded, decoded and interchanged through heterogeneous platforms. In such cases, HyTime which is more suitable for the document authoring and publishing applications than the real-time interactive applications, should not be exploited as the basic standard for the information processing. On the other hand, MHEG specifies the information interchange format as object instead of the more limited concept- document as in HyTime. An object to be interchanged in an MHEG system can be a document, a media object, or even a part of a document. This adds to the reusability of the courseware and the flexibility in rendering the courseware interactively. Therefore, MHEG is chosen for the information delivery format in our TeleLearning system.

From the technical point of view, some advantages of exploiting MHEG as the information delivery scheme in our system are:

**Object-oriented Approach:** The MHEG model has made use of some object-oriented concepts such as object, class, inheritance and overridden. This has enable autonomous and reusable information structures. Information is coded into MHEG objects, stored or interchanged, and finally presented to the end-user. Run-time objects, which are created from an interchanged model object, can provide more reusability and save transmission resources.

**Interactive Multimedia and Hypermedia Information Coding:** MHEG has provided structures for the composition of multiple media with relationships among them into a single unit of interchange. The object designers can specify interactions through link class and action class, and provide them to the end-users through a Graphical User Interface (GUI). Hypermedia links are also supported by link class.

**Minimal Resources:** Information of resources required to present the encoded data can be coded into a descriptor object and transmitted to the presentation environment before the real content objects are transmitted. This can facilitate a correspondence between the resources required to present the objects and the resources available to the system. Descriptor objects can also perform a negotiation between the source of the MHEG objects and the presentation environment.

## 3.2 Generic Architecture

MITS is a distributed multimedia system. It is composed of five basic components distinguished by the services they provide (Figure 3.1).

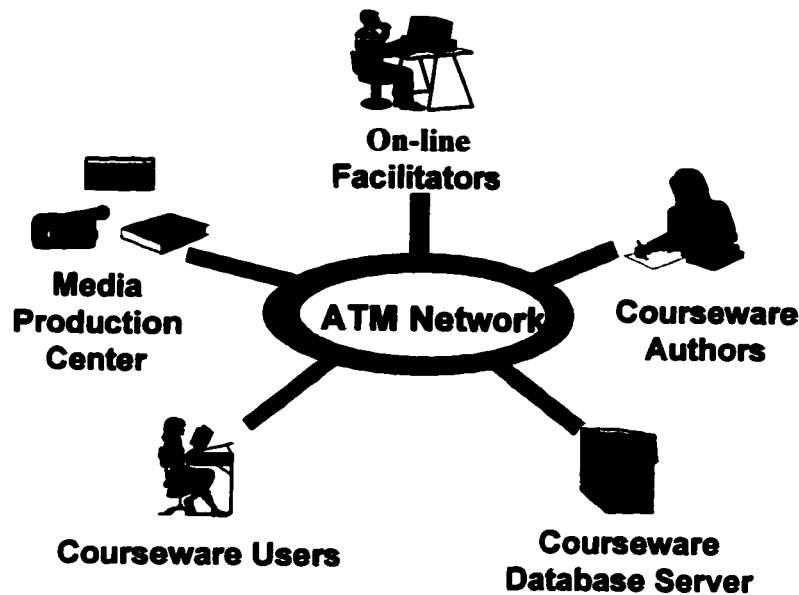


Figure 3.1 A generic architecture for MITS

A *media production center* is responsible for capturing information from the real world and coding them into different media objects such as text, image, audio, and video. These media objects are stored in the database and used as basic material for courseware authoring. *Author sites* are distributed over the network for authors to access the database and construct courseware. The authoring of a courseware usually includes two aspects. First, content of the course is specified by directly creating media objects or making use of the objects retrieved from the database. Second, the scenario and the layout of the presentation are defined by specifying the relationships among content objects in a courseware. A courseware is also stored in the database after being created, and it can be

### Chapter 3. A Broadband Multimedia TeleLearning System

---

updated in both the content and the scenario at anytime. *Courseware database* for storing courseware is the third basic component in the system. It consists of a content server and an object-oriented database. In whatever way the database is structured, users are hidden from the details of data operation. Query or presentation services are offered in a fast and convenient manner. The *courseware navigator* at each user site handles the access to the courseware stored in the database in accordance with pre-defined scenario or user interactions. Through a well-designed GUI, it provides various kinds of learning services to the students in a seamless integrated environment. When a student encounters a problem during learning, he can always get help from an on-line facilitator. Teachers or specialists will work on-line to answer questions, or discuss interesting topics with the students on their request.

All these components are distributed over a computer network and work together to offer an interactive multimedia courseware service. The scale of the network, the number of each basic component should be chosen carefully according to the real-world requirements as well as the network capability so as to offer a high performance-to-cost ratio system.

### 3.3 An MHEG-based Model for Information Interchange

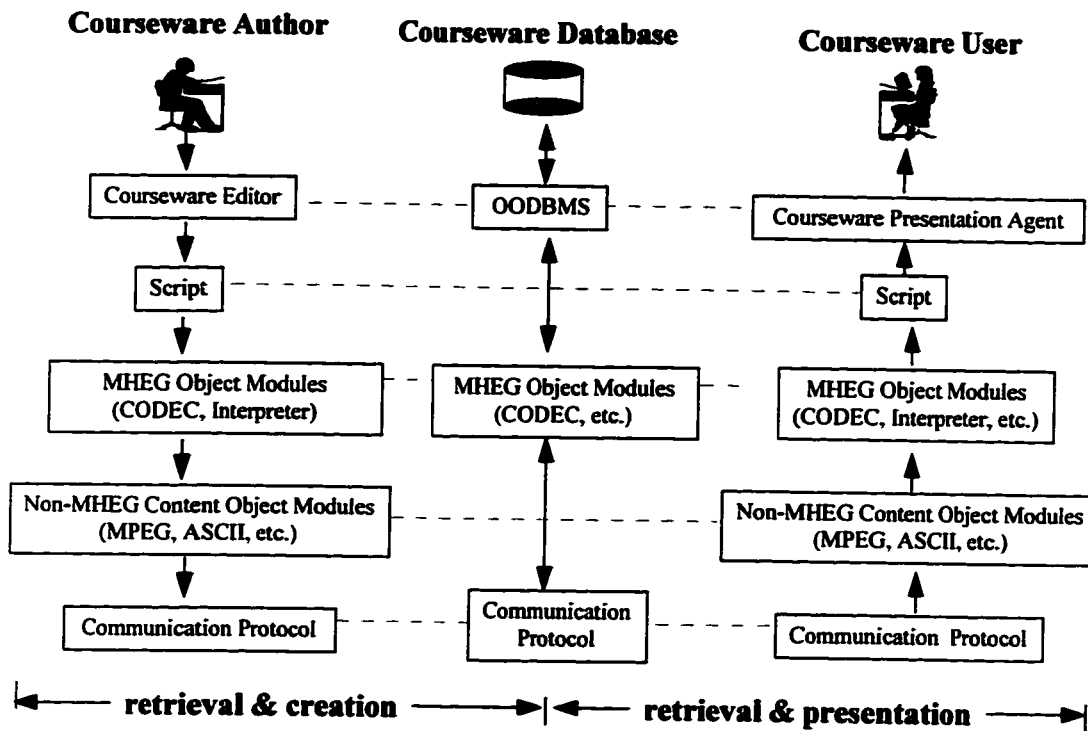


Figure 3.2 An MHEG-based model for MITS

The layered model for interactive multimedia courseware delivery shown in Figure 3.2 is designed based on MHEG standard. Three sites are involved in the knowledge transmission process: authoring, storage, and presentation. Basically, all the layers in the author site and the presentation site are symmetrical. For example, in the MHEG layer, the author site encodes the objects into ASN.1 [MHEG, 95] or SGML [MHEG, 95] format for interchange, and the user site decodes this interchange format into its own internal format for further interpretation and presentation. The knowledge stored at the database site can be viewed as an intermediate status in the information

interchange process. MHEG objects are received and decoded into the internal object format to be stored and operated by the database management system. When requested by an application, the objects are encoded again into MHEG format and transmitted to the application. Therefore, only communication layer, MHEG layer, and application layer are kept for data processing at this site.

The application layer includes a courseware editor, a courseware database and a courseware navigator, which are located at each site of the system respectively. The script layer defines complex relationships between MHEG objects and run-time objects for the courseware presentation. This exchange may make use of the MHEG object interchange standard at the next lower layer. In the MHEG object layer, objects are coded into ASN.1 or SGML at the courseware author site and transmitted through the network. When the objects are received by the courseware navigator, they are decoded and interpreted for playback. This layer also handles the presentation process such as resolution of object references, creation of runtime-objects, as well as interpretation of links and actions. The non-MHEG content object layer provides mechanisms to handle differently coded media objects. Content object indicates those mono-media objects such as video, audio, text, etc. They are originally produced at the media production center using media coding standards such as MPEG, JPEG, ASCII depending on the types of the media. The communication protocol layer is responsible for exchanging the messages and acknowledgments required by the applications.

For an interactive multimedia learning environment, it is not required a specific communication network to be used. However, for obtaining good quality of service in

real time presentation of dynamic media such as video and audio, we suggest broadband network to be chosen for the communication infrastructure for this application.

### 3.4 Courseware Processing: Models and Strategies

The whole life cycle of a courseware composed of three phases: production, storage and presentation (Figure 3.3). The modules for processing the courseware at each site are shown in Figure 3.4. A using application, a user interface, an MHEG engine and a Communication Management Module are common modules installed at every site.

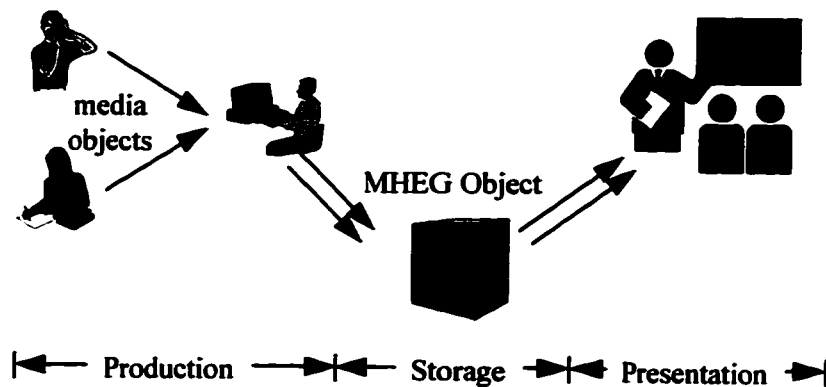


Figure 3.3 Processing model for multimedia courseware

The *User Interface and Presentation Service* is designed to handle the look and feel of the user interface of the courseware applications- more precisely, the courseware editor and the courseware navigator. The *Courseware Editor* provides utilities for the courseware authors to specify courseware scenario, describe courseware presentation layout, as well as create and modify content objects. The *Courseware navigator* is a using application to play back the courseware at the courseware presentation site. It retrieves MHEG objects from the courseware database according to a pre-defined scenario or the user interaction. The *Database Management System* manages the MHEG objects stored in

the courseware database. Functionalities such as adding, deleting, updating and retrieving are provided. The *MHEG Engine* is a set of software modules designed and implemented by the system designer to encode, decode, handle or interpret the MHEG objects. Certain sub-modules in it may or may not be used at specific site in the environment. For example, in data storage site, it is not necessary to have an MHEG interpreter because no object is going to be presented directly at this site.

### **3.4.1 Courseware Production**

Courseware production involves the creation and edition of both the content objects and the rendering scenario for the presentation. By using video and audio capturing devices such as video cameras, microphones, and PC-VCRs, the media production server provides all the data needed for the creation of a multimedia courseware. Support is also available for editing of graphic and text objects. Media objects are stored into the content database after being created at the media production center.

A courseware is created at an *Author Site*. Under a teaching architecture suitable for presenting the course, an author creates new content objects or chooses the existing objects from the content database, then specifies the rendering scenario for the courseware. Interactive and complex courseware presentation can be achieved by using several MHEG object classes such as composite class, link class, action class, script class, etc. Details of the courseware authoring will be discussed later in chapter 4.

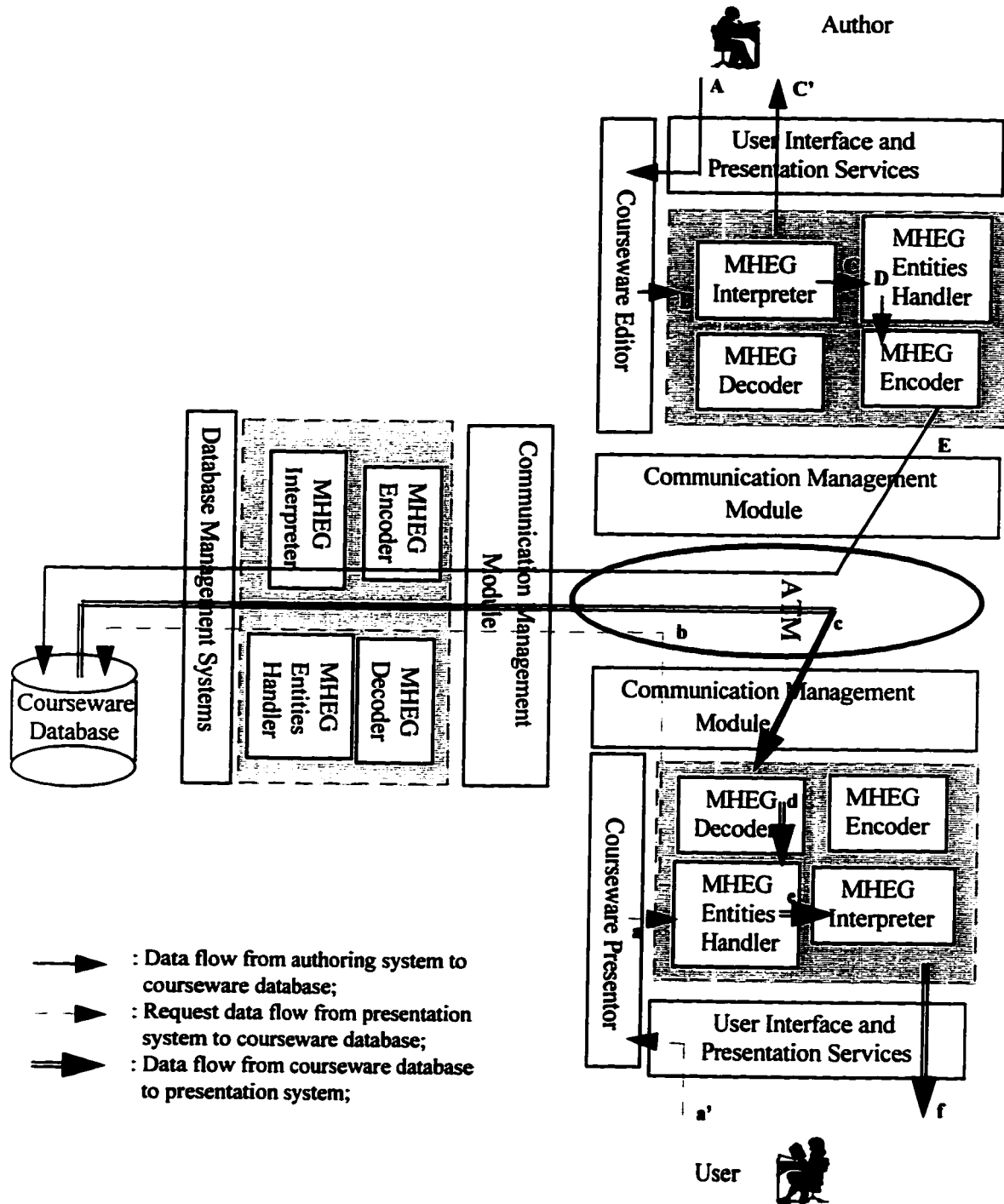


Figure 3.4 Courseware processing modules in MITS

### 3.4.2 Courseware Storage and Retrieval

The *courseware database* is a large, distributed, object-oriented, multimedia database. It stores all the MHEG objects as well as the content data of these objects. In the coding schemes of the MHEG objects, content data of different media types could be either included directly as binary data in an object, or stored separately in a content database and referenced by MHEG objects. In MITS, the latter scheme is chosen and the content data is stored separately from the scenario so that reusability of the content objects is achieved among different applications. Moreover, while content objects of large size are transmitted only at the time they are requested, the transmission resource is saved and the real time performance is improved. The communication model for database and the end-user is a client-server model (Figure 3.5). This architecture is designed to ensure transparency in the interchange of information so as to screen end-users from the underlying architecture of the application. Distributed applications complying to the model may consist of a number of independent programs running on remote hosts scattered over the network, without the users' knowledge [Ben, 95]. A database server waits and listens for a service request from a client. When such a request is received, the server retrieves objects in the database according to the information provided by the client. Then it establishes connections to the client and transmits the MHEG objects or the content data through the network to the end-user. Distributed over the ATM network, user sites all have client routines installed in the software so that browsing and queries on the database may be performed.

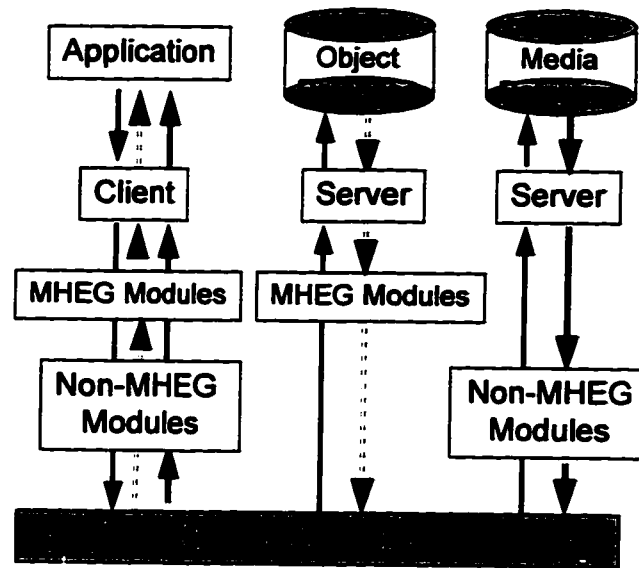


Figure 3.5. Client-server model in MITS

### 3.4.3 Courseware Presentation

At the *Courseware Presentation Site*, courseware navigator controls the presentation process according to a scenario pre-defined by an author. Meanwhile it handles the users' interaction through a GUI. On receiving the request, the MHEG engine resolves the MHEG object references, and then goes to the courseware database locally or remotely to retrieve the MHEG objects. As a result, encoded MHEG objects are sent to the MHEG engine at the courseware presentation site. After being processed by the modules in each layer, the object is presented to the end-user. Apart from the courseware presentation, the courseware navigator may also provide features such as the library browsing and the communication facilities for discussing.

## Chapter 4

# Courseware Authoring in MITS

Authoring a multimedia courseware involves the process of not only capturing information from the knowledge resources, but also structuring them into a semantically coherent document for knowledge diffusion. In a traditional textbook, knowledge is structured in a linear mode and is relatively easy to be authored, while authoring a multimedia courseware in a non-linear structure is much more complicated. Moreover, in MITS where learners are the controllers of the learning process, teachers can only affect the learners indirectly by structuring the course presentation according to the learning theories. In another word, courseware authoring is the only step during which a teacher can affect the learning process and guide the students to acquire knowledge in a proper way. Therefore, it is the key step in a courseware life cycle. Whether or not a courseware is successful is often decided at this step.

## 4.1 Multimedia Courseware Production

### 4.1.1 General Process for Multimedia Courseware Production

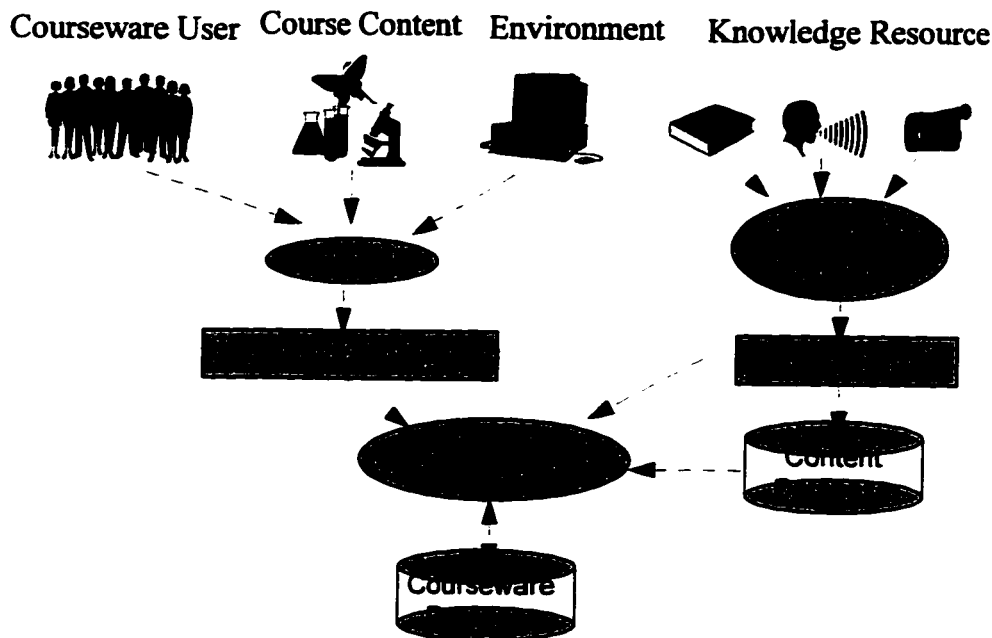


Figure 4.1 General process for multimedia courseware production

Multimedia courseware production usually involves two roles: a media producer who captures information and produces the course material, and a courseware author who constructs the course material into a courseware guided by education theories. What we are more interested in is the courseware authoring. The following steps are usually involved in authoring a multimedia courseware (Figure 4.1). *Analysis* is the first step in which the type of knowledge transmitted, the kind of knowledge acquirers involved, and the resources available are studied. As the result, a proper *teaching architecture* [Schank, 90][Schank, 94] for the courseware is chosen. Some examples of the teaching architectures can be simulation-based model, case-based model, etc. The teacher then

authors the courseware by semantically integrating the media objects into a teaching architecture he has chosen. The courseware produced is stored in the courseware database.

### 4.1.2 Courseware Authoring in MITS

The courseware authoring in MITS is done in four layers with different degree of complexity: teaching architecture layer, document layer, object layer, and media layer (Figure 4.2). Each layer provides service for the next higher layer. Mechanisms for mapping the specification from the higher layer to the next lower layer are to be provided by the courseware-authoring environment.

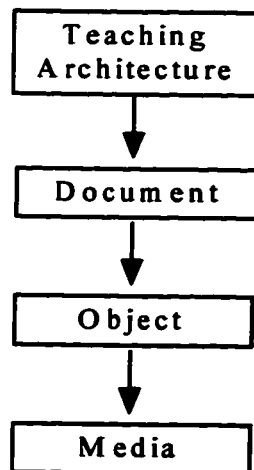


Figure 4.2 Layers for multimedia courseware authoring in MITS

The *media layer* is implemented at the media production center. It provides courseware material in different kinds of format, e.g., video, audio, text, graphics, image, etc. The *object-authoring layer* is implemented for the instantiation of the objects from the MHEG classes. The objects can be content objects which include references to the

media objects and the presentation parameters. For example, a video object can be instantiated from the MHEG content class with the following attributes specified:

Media object = "Paris.mpg";

Coding method = MPEG;

Size = 64\*128;

Number of frame = 180;

Position = (100, 200);

There can also be synchronization objects for scenario specifications. Courseware authors are usually educators who are not required to have any knowledge of multimedia data format. Technical details such as MHEG data coding should be hidden from them. For complicated multimedia presentation in a course, it is too demanding and error-prone to specify all the scenario just in the object layer. Therefore, a multimedia *document model* should be chosen to facilitate the organization of the MHEG objects. The *teaching architecture layer* is a layer for the courseware authors to choose appropriate teaching models for the course presentation. Different teaching architectures may result in different document models for the courseware authoring.

## **4.2 Teaching Architectures**

In [Schank, 94], Dr. Roger C. Schank has proposed six teaching architectures which combine the learning theory with the modern computer capabilities to provide the true one-on-one based teaching. Sample courseware has also been developed to demonstrate these models. These teaching architectures are introduced as follows.

### **Simulation-based Learning by Doing**

Experience is the best teacher. Children learn walking by doing, failing and doing again. Of course, during this course, hearing about or copying some new behavior that has better result is also very helpful. The simulation-based architecture is usually used when apprenticeship will not work all that easily, or is risky, e.g., pilot training. It is composed of four parts: a simulator, a teaching program, a language understanding program, and a story-telling program. The teaching program helps the student through the simulator and discusses issues with the student. The language understanding program interprets the student's questions to languages that computer can understand. The story-telling program is activated by the teacher at appropriate time to tell stories from the experiences of experts in actual situation [Schank, 90].

### **Incidental Learning Architecture**

The basic principle of this architecture is that when doing something fun, the students can learn a great deal without noticing it. The Institute of Learning Science (ILS) at the Northwestern University has built a software called "Road Trip" for the geography course. In this program, students are given a simulated car and allowed to tour around the United States. Upon arrival at a destination, the student can watch exciting video clips spotlighting activities or events in that location [Schank, 94].

### **Learning by Reflection**

Sometimes a student does not need to be told something, but rather needs to know how to ask the right questions. A student can be her own best teacher if she just has someone around to listen to the ideas she generates. The teacher's role in this architecture

is to help the students see shortcomings in thinking. A sample software called Movie Reader has been developed using this architecture [Schank, 94].

### **Case-based Teaching Architecture**

This architecture depends on two ideas: experts are repositories of cases, and good teachers are good storytellers. The students are told exactly what they need to know, when they need to know it. When students realize that they need information to progress, they will learn fast. The case-based architecture can be combined with the simulation-based learning by doing architecture. The learning by doing architecture provides the activity, and the case-based architecture provides the instruction. A biology tutor called "Creanimate" was developed at the Northwestern University that supports these architectures. Creanimate invites children to create their own animal by taking an existing animal and changing it in some way. Then it responds to the created animals as a naturalist encountering them for the first time in the wild and raises open-ended questions, discusses possible answers with a student.

### **Learning by Exploring**

An important method of teaching is to answer a student's questions at the time he generates them. Learning by exploring architecture provides this environment to the students. In this architecture, students must be free to follow their own path, and they should have multiple experts available to answer their questions.

### **Goal-directed Learning**

Providing a goal that the students will adopt willingly, this architecture can leverage the power of the teaching architecture. Also provided is a way for the students to control the environment in which they learn, and an opportunity to adapt what is presented to them to their existing needs. A project developed by ILS called “Museum visitors as genetic counselors” showed good effects of this method.

## **4.3 Interactive Document Models**

It has been proved that non-linear knowledge structure is superior to the linear structure for education and especially, for training. During the past few years, several non-linear knowledge structures have been actively exploited in the TeleLearning applications.

*Hypertext* represents text-based knowledge as a semantic *net* or a *web*, where semantically related knowledge nodes are linked and can be traversed through *hyperlinks* [Midoro, 93]. While reading through the text, a user can always follow the pre-constructed hyperlinks to find out more information about a concept or an expression which attracts his attention. In another word, the knowledge is modeled as a web of concepts or ideas instead of a series of linearly presented concepts. However, without the capability of processing multimedia, hypertext is still not good for lots of education applications. *Hypermedia*, as the union of hypertext and multimedia, has the same knowledge structure except that links may point to media objects of any types including video, audio, and animation. Hypertext and hypermedia applications provide the non-

linear knowledge structures while still resemble the traditional text-based knowledge diffusion mode in its interface. Therefore, they have a reinforcement effect [Grace, 95] than the traditional text-based applications.

As a novel and more natural way for people to get knowledge in a TeleLearning system, interactive applications have attracted more and more attentions from the educators.

### **4.3.1 Static Interaction and Dynamic Interaction**

An interactive multimedia document can provide one of the following two interaction styles: static interaction and dynamic interaction.

*Static interaction* indicates the behavior of those documents which are statically formatted without the time synchronization. The playback of such a document is driven only by the users' interaction (e.g., browsing or querying). The above mentioned *hypermedia model* is in this category. However, since no pre-defined scenario is provided in the knowledge structure, educators cannot exert any influence on the learning process. The navigation in the knowledge space is driven only by the learner. It is a very common experience for most of the learners to get lost while navigating in a large-scale web. Therefore, the use of such knowledge structure should be limited only to the sophisticated knowledge acquirers who know exactly what to learn and how to filter knowledge based on their experiences. A lot of efforts have been made on solving the problem of getting lost by optimizing the design of the hypermedia system. One example is the

Dexter model [Dexter, 94] where a graph of nodes and links is defined and different views of the graph can be generated according to the user-specified criteria [Rada, 96].

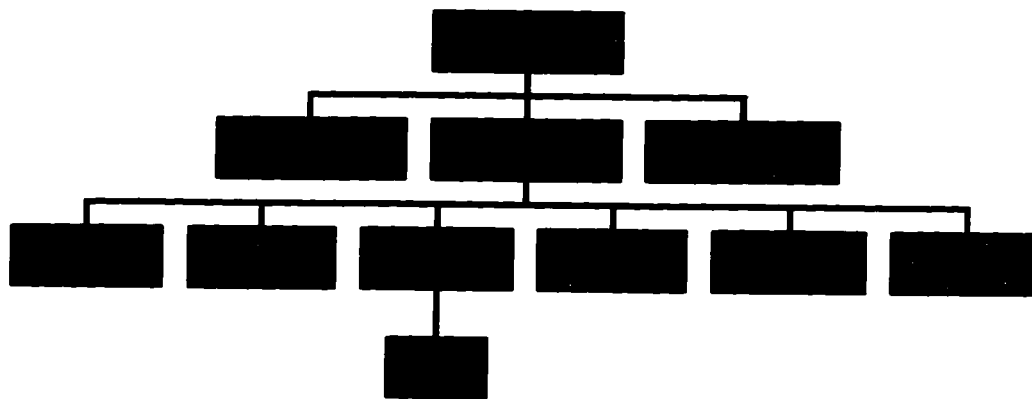
A *dynamic interactive* document has both pre-defined rendering scenario and interactive user-interface. Therefore, the playback of such documents can be driven either by the end-user, or its own scenario in case that the learner has chosen to watch the presentation passively. We have designed an *Interactive Multimedia Document Model* for this kind of application. The model organizes knowledge in accord with a teaching architecture and leads the students through the learning process by its presentation. Meanwhile it still provides hypermedia-typed choices in its interface that allows users to interact with the presentation. In this way, students can learn under a clear goal, and in a relatively focused domain without being worried about getting lost. On the other hand, they can still control the learning speed and tailor the material to fit for their needs. Dynamic interaction provides two-way interaction between the learner and the courseware presentation. It is closer to people's natural learning process than other knowledge transmission methods, and theoretically should have better educational effects, especially for non-sophisticated learners. However, effectiveness of such application is also subject to the type of knowledge transmitted and the profile of the knowledge acquirers involved [Grace, 95].

Based on the requirements of the modern teaching architectures and the emerging document models for courseware, we have proposed one document model for each interaction style for courseware authoring in MITS.

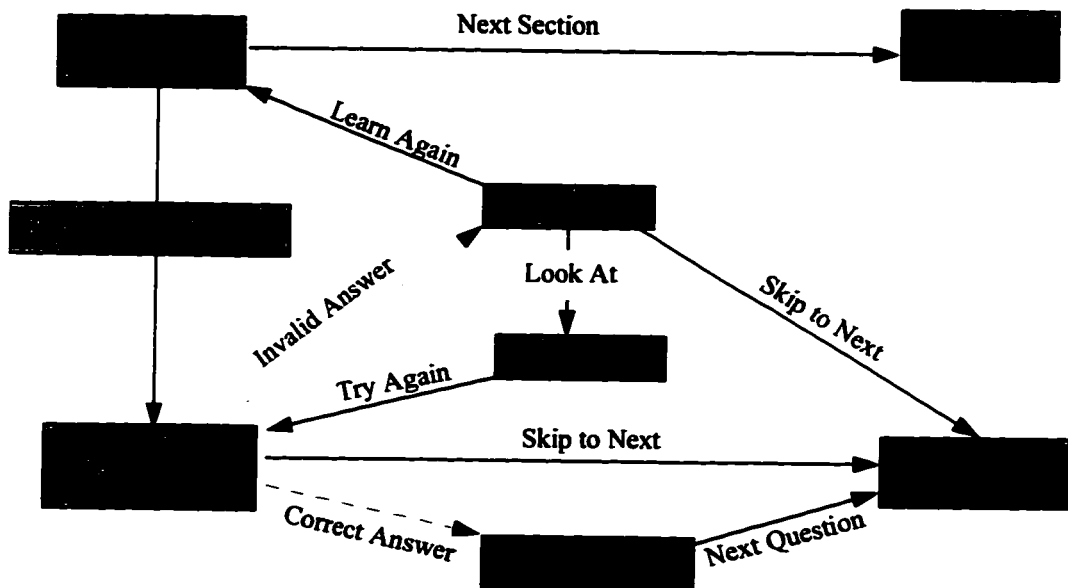
### 4.3.2 A Hypermedia Document Model

A hypermedia document structures information as nodes and links. A node is usually laid out as a page or a word in a page in a document, while a link is the path from one node leading to another. Usually links are specified between semantically related nodes so that a user can easily follow the links and find out a bunch of related information immediately as he requests. A hypermedia document is modeled with a logical structure, a layout structure and a navigation structure.

The *logical structure* describes the organization of the content object in a multimedia document (Figure 4.3a). A document is composed of a number of pages, and each page may contain many media objects. For interactive behaviors, “choice” is added as a new media object. Word is the smallest component in the logical structure which is usually specified as the source of a link. The *layout structure* specifies the spatial characteristics of different media objects in a page. The *navigation structure* defines the hyperlinks between certain two logical nodes for the presentation application to follow in responding to the users' interaction (Figure 4.3b). It has nodes, links as well as the conditions to activate the links. Conditions are usually buttons or special clickable text in layout of the document.



(a) Logical structure



(b) Navigation structure [Jocekyne, 95]

Figure 4.3 Hypermedia document model

Figure 4.3b is part of the navigation structure designed for a hypermedia document. The current section of the document will link to the next section of the document when "Next Section" is chosen. While "Test Your Knowledge" is chosen, it will lead to a question to be presented. A group of questions is provided for testing the

learning efficiency. Different response from the learner will lead to the different nodes to be presented.

### **4.3.3 An Interactive Multimedia Document Model**

An interactive multimedia document is more complicated than the hypermedia document. On one hand, inter-media temporal relationships should be specified. On the other hand, the hyperlinks are more complicated which may include not only the click of word or buttons, but also status changes of dynamic media objects as conditions to fire a link. In one word, the scenario can be more complex. We describe this model using a logical structure, a layout structure, as well as a rendering scenario including a time-line structure and a behavior structure.

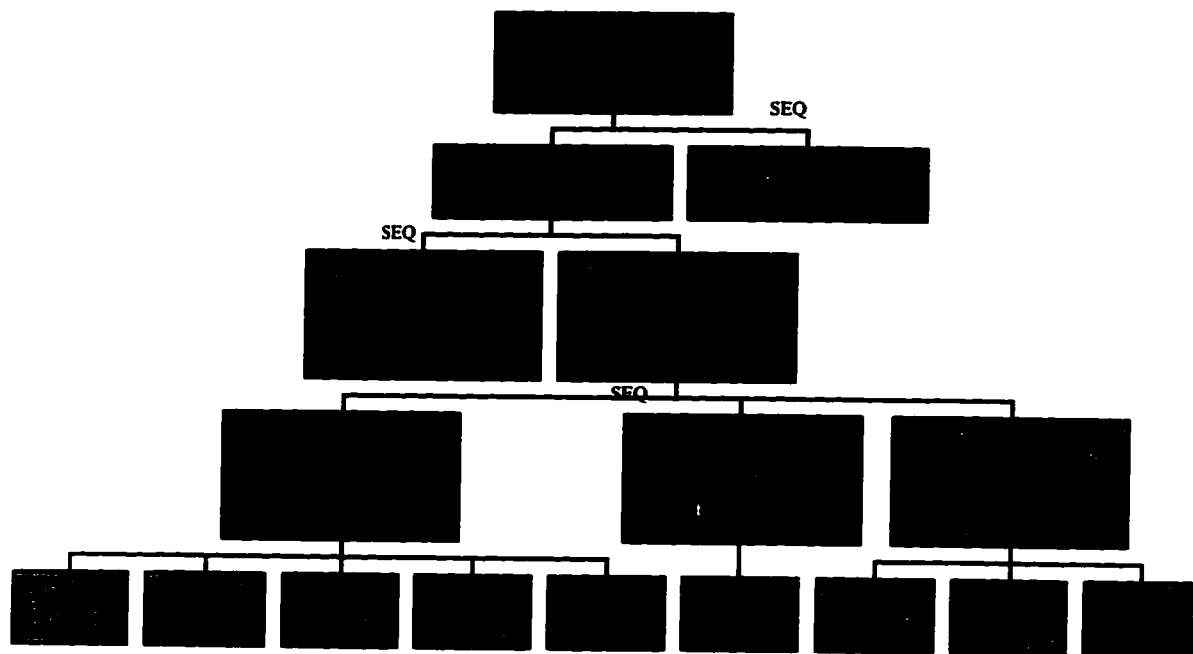
The *logical structure* is designed based on the MHEG composition concept. An interactive multimedia document can be repeatedly divided into sections and subsections until a scene. A *scene* is the grouping of a certain number of objects presented in the same space for a certain period of time (Figure 4.4a). Sections, subsections, as well as scenes are composite MHEG objects with different degrees of composition. All the divisions are decided according to the logical integrity. The presentation sequence of these composite objects is pre-defined to be simple serial playback when there is no users' interference. It should be noted that we use the term "scene" in place of "page" which is used in a hypermedia model. The difference of these two terms is that there may be dynamic media objects (e.g., video) presented in a scene with pre-defined temporal characteristics. We consider a scene to be a wider concept than a page. For example, one scene may be video

clips showing different halls in a museum with synchronized audio to introduce what is being shown, while buttons of "stop", "show caption", "enter hall", are provided which can redirect or interrupt the playback on the screen. The media objects in a specific scene usually have complex relationships in time, spatial, and even behavior. The layout structure and the rendering scenario are designed to specify these relationships.

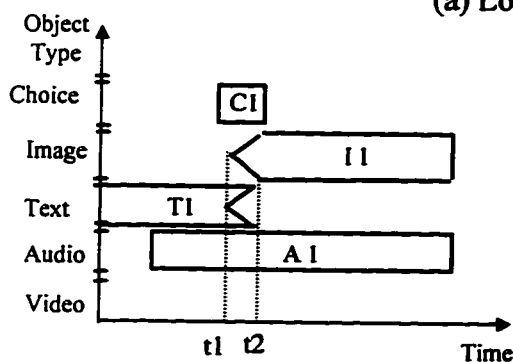
The *layout structure* defines the presentation characteristics of the user perceptible objects in a document. For visible objects such as video, image, graphics, and text, some important parameters should be specified including the coding method, position, and size of the object to be shown on the screen. Generic values or special terms can be used. Some special terms may be "in front of", "behind", "on top of", "below", "to the left", "to the right", etc. For audible objects, parameters such as coding method, sample rate, volume, tune, direction (e.g., forward, reverse) and duration are to be specified. To make use of the runtime-object mechanism provided by the MHEG standard [MHEG, 95], a channel is to be assigned to a runtime-object. As we know, runtime-objects are copies of model objects which are only created, manipulated, and finally deleted in an MHEG engine at the presentation environment. A channel is a logical space in which the runtime-components are positioned, presented and perceived by the user when they are mapped to the physical space. The mapping is handled by the MHEG engine, while the channel should be defined and assigned for a runtime-object in the layout structure by the object designer.

A time-line structure, as part of a rendering scenario is used to specify the temporal relationship of the media objects to be presented (Figure 4b) in a scene. The temporal model for interactive multimedia scenarios proposed in [Nael, 95] is exploited.

In Figure 4.4b, choice1 is a button object provided for the users to choose changing from viewing text1 to image1. Pre-defined duration is specified for text1. Without the user's interaction, text1 will be changed to image1 at time t2. However, users can click the button 'choice1' at any time between t1 and t2 to display image1 earlier than the pre-defined time. Therefore, the playback time of image1 is dynamic. This is a simple example of the time-line structure for dynamic interaction.



(a) Logical structure



(b) Time-line structure

Condition		Behavior	
Object	Status	Object	Action
Stop button	Selected	1. Audio1 2. Text1 3. Image1	Stop
Text1	Stop	1. Image1	Run

(c) Behavior structure

Figure 4.4 Interactive multimedia document model with an ATM course as an example

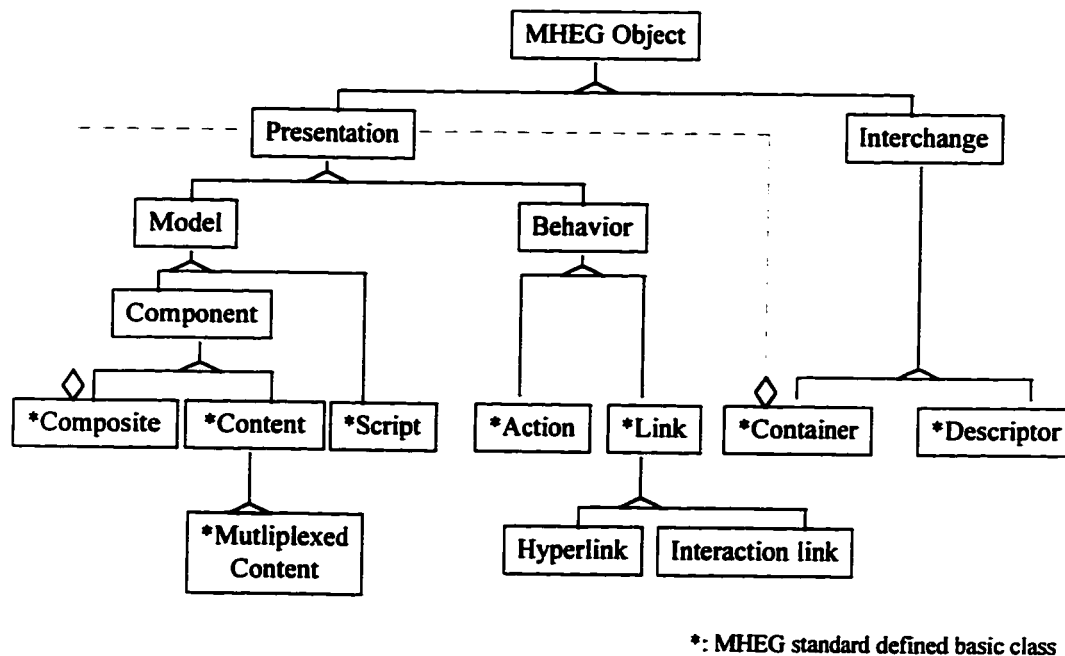
A behavior structure, as another part of the rendering scenario specifies special links between the media objects or between users' action and the media objects. It is composed of a set of conditions and a set of actions to be activated while the conditions are met. A condition can be a user input or a status change of a media object. A condition set is composed of trigger conditions and additional conditions. When the MHEG engine detects a change in the value of an object status or a presentable status that has met the trigger condition, it continues to check if additional conditions are met. When all the conditions are met, the link is fired. The actions to be applied on the media objects, as the result of the fire of links, are defined according to the processing capability of the MHEG engine. Some examples can be "start", "stop", "prepare" of certain media objects. Figure 4.4c shows two examples of defined behavior. In one example, when user has clicked a "stop" button on the screen, `audiol`, `text1` and `image1` is going to stop their playback. In another example, when `text1` stops being displayed, `image1` is to be shown.

### **4.4 MHEG Object Authoring**

MHEG objects are categorized by classes, where the objects in a class share behavior, characteristics and functions which manipulate them. As described in section 2.2.2.1, eight basic MHEG classes are designed for general applications. A given object is an *instance* of a particular class. Each class has a list of *attributes* associated with it which govern the behavior of objects in that class. The design of MHEG object classes relies on the analysis of their common behavior and the commonality of properties between object categories. This can lead to a single or multiple inheritance scheme. On

studying the MHEG standard, a basic MHEG class library for multimedia and hypermedia information coding is designed. Meanwhile, for the courseware authoring in MITS, we have also designed the courseware class library. Further studies and application requirements may lead to new classes to be added into the basic class library or the extended class library for the courseware application.

### 4.4.1 An MHEG Class Library

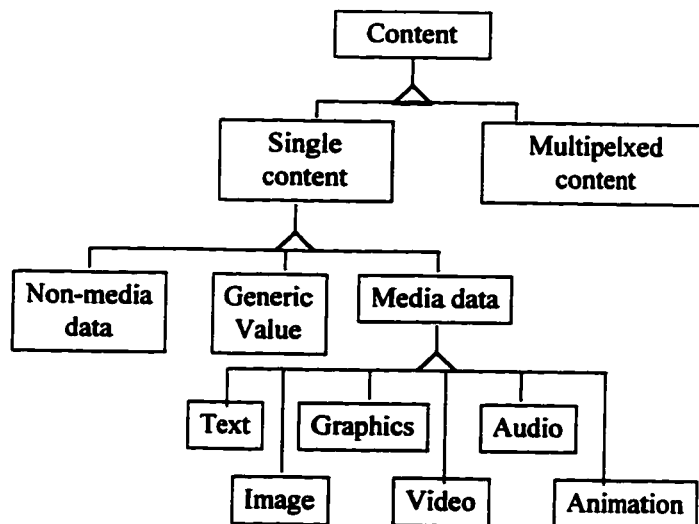


(a) Class hierarchy chart for basic MHEG defined classes

Figure 4.5 A basic MHEG class library

Figure 4.5 shows an MHEG class library designed based on the MHEG standard. Extensions are derived from the eight basic MHEG classes defined in the standard to provide more practical and detailed object instantiation. Basically, MHEG objects are categorized into two abstract classes: presentation class and interchange class. Any

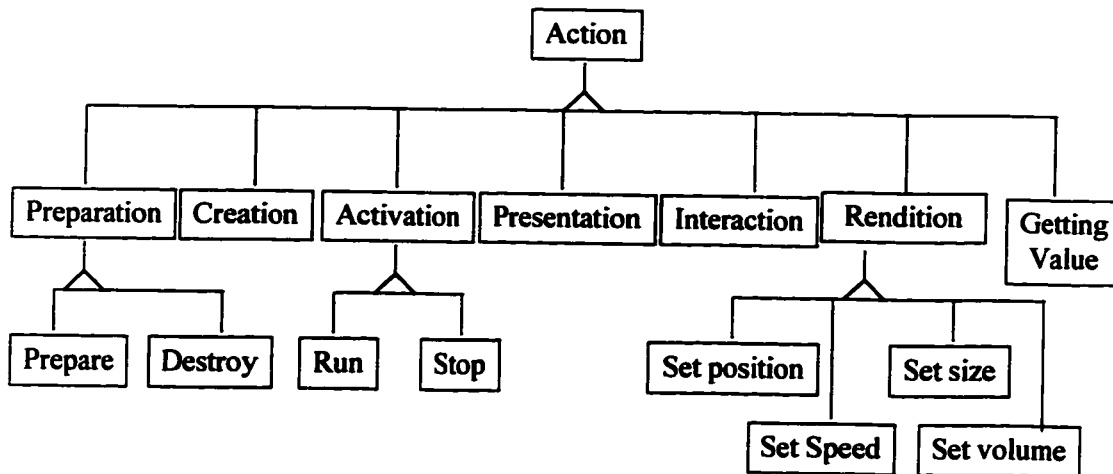
subclass of the presentation class can be aggregated into a composite class for presentation, or a container class for interchange. From a model object which is instantiated from a model class, any number of run-time objects may be created based on the instructions given by an object author. The activation of a runtime-object does not affect the model object, which allows the reuse of the same model object in different context. The model class is an abstract class inherited by script class and component class. The component class is an abstract class inherited by content class and composite class. All the other eight classes with asterisks are the classes defined by the MHEG standard. Common attributes of the MHEG class are identification of the standard and standard version, identification of the class of the MHEG object, MHEG identifier of the MHEG object, and general object information (e.g., name, owner, version, date, keywords, copyright, license and comments). Each class also has its own attributes defined. Details can be found in [MHEG, 95] section 16-27.



(b) Class Hierarchy chart for content class

Figure 4.5 A basic MHEG class library

The subclasses of the content class are derived for different kinds of content objects. The standard identifier attribute "19" which stands for "MHEG" in MHEG objects will not be inherited by the single content object class. Three subclasses are derived from the single content class: media data class, non-media data class, and generic value class. The media class includes differently coded media types (e.g., video) as its subclasses. The non-media data class objects may be executables or document coded in other formats (e.g., HyperODA, HyTime). In the generic value class, a value may be stored in the data for a comparison, an assignment or a presentation. The multiplexed content class, as another subclass inherits all the attributes of the content class, while adding data with a description for each multiplexed stream. A stream identifier encoded as an integer can be used to control single streams, for example, to turn audio on and off in an MPEG system stream [Thomas, 95].



(c) Class hierarchy chart for action class

Figure 4.5 A basic MHEG class library

Objects of an action class are used to control the behavior of objects. Several subclasses are derived for different kind of controls. *Preparation* controls the availability

of the object in the system. *Creation* builds presentation instances from a component object and script instances from a script object. *Presentation* controls the progress of the presentation instances in the system. *Activation* controls the activation of the script instances in the system. *Interaction* determines the results of the interaction between a presentation instance and the system. *Getting Value* gets the attribute, the status, or the behavior value of an MHEG object, presentation instance, or script instance which express the link condition in a link object. *Rendition* prepares the rendition of the presentation instance on the system according to the media type. For example, use "Set Speed" for the time-based media and "Set Size" for the visible media.

#### 4.4.2 A Courseware Class Library

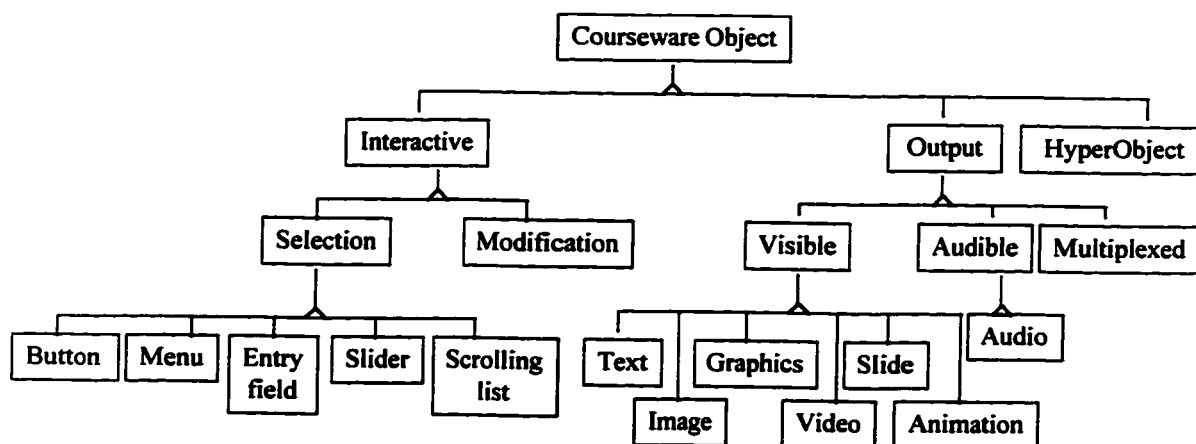


Figure 4.6 An interactive multimedia courseware class library

The basic MHEG library is designed based on the data requirements of general applications. However, it is the potential of the MHEG coding format to provide as many composite objects for different domains of applications. The composite object can only be designed according to the real application requirements. For the convenience of

authoring a courseware, we have analyzed the most frequently used object classes and their coding requirements. A courseware class library is built upon the basic MHEG class library so that courseware authors can easily create objects by instantiating them directly without any deep understanding of the MHEG concepts. In fact, this library acts as a bridge between the courseware authors and the MHEG coding format.

There are three courseware object types: Interactive, Output, and Hyperobject.

- **Interactive object:** Composite objects which represent styles of selections in GUI (e.g., buttons, menu, entry fields, etc.), input from the users (e.g., buttons activated, menu selections, input to forms or input to edit fields), as well as the resulted actions they may lead to.
- **Output objects:** Basic or composite object that is intended to be presented in some way to the user (e.g., text, graphics, images, audio or audiovisual sequence).
- **Hyperobject:** Composite object which consists of input and output objects plus explicit links between them.

## **4.5 Implementation Issues for a Courseware Editor in MITS**

The main concern of a courseware editor in our system is to give courseware authors more power and flexibility to implement their education theories, while hiding the details of the MHEG object coding. The layered model for courseware authoring was designed under this concern. However, it should be achieved through feature selection and user interface design of the courseware editor. The courseware editor is responsible

for the mapping between layers in the authoring model. Framework and template are provided as facilities for more convenient authoring.

### **4.5.1 Framework**

Several models for teaching architecture are to be provided to the authors in the forms of frameworks. As described in section 4.1, the courseware authors first find out the most appropriate teaching architecture for presenting the course, and then use the courseware editor to author the courseware. Therefore, the first step is to provide frameworks according to the teaching architectures. The chosen of a specific framework will result in a corresponding document model to be selected. The views related to the document models are going to be shown on the screen. The courseware authors need only to fill the media objects into the frameworks and specify the scenario for the presentation.

### **4.5.2 Template**

As described in 4.3.2, a courseware class library is designed for the most frequently used courseware object classes. In the courseware editor, these are provided in the form of template for the creation of the courseware objects. For example, a template for a video object can have parameters such as position, size, duration, playback speed, and links. Courseware authors just need to specify references to the media objects, temporal, layout and behavior parameters through a friendly user-interface. The detailed MHEG coding mechanisms are hidden.

### **4.5.3 Issues for GUI Design**

An interactive multimedia courseware authoring environment includes a logical view, a layout view, a time-line view, as well as a behavior view for the courseware or scenes in the courseware being created. The four views are the implementations of the logical structure, the layout structure, the time-line structure, and the behavior structure which was discussed in section 4.2.2. A logical view shows the hierarchical structure of a document. To simplify the work of authors, layout view is provided not only for them to preview the courseware layout, but also to specify the spatial parameters and spatial relationships in a more intuitive way. For example, specifying position of a video object on the screen can be done by selecting the object from the logical view, and then dragging and dropping it to the proper position in the layout view. A time-line view is designed for authors to view and to adjust the time relationships in a graphical way. However, in specifying the temporal parameters or relationships, it would be more straight and simple to choose special terms such as "before", "after", "meet", etc. [Petra, 92], and then assign the generic values. These are provided through a dialog box. The behavior view shows on the screen as a table with two fields. One is the condition set, and the other field is the action set. The user interface allows not only displaying, but also adding and removing items from the set.

The authoring environment for a hypermedia document model includes a page list, a page view, and a navigation view. The page list shows the title of all the pages as well as the media objects included in each page. The page view shows the layout of the media object on a page. The navigation view, as the interface of a navigation structure for

## Chapter 4. Courseware Authoring in MITS

---

the document can provide a subset view of the navigation structure to show all the nodes which are linked to a specific node. By clicking any node displayed in the current subset navigation view, a new subset of the whole graph with the clicked node as the source node is going to be shown. Authors are allowed to add, remove or browse the content of any node in the graph.

## Chapter 5

# **Implementation of a Courseware Navigator in a Broadband Multimedia TeleLearning System**

So far, we have discussed the system architecture as well as the courseware production of a broadband interactive multimedia TeleLearning system. Considering the stability of the MHEG standard as well as the availability of the MHEG object coding system, we have decided to implement our first prototype on the MEDIABASE platform. The MEDIABASE platform is a multimedia information and communication system which has already been under development in our lab.

## **5.1 A Broadband Multimedia TeleLearning System**

In our Multimedia Information Research Lab, a MEDIABASE project has been under development for years. It is a multimedia information and communication system with a particular focus on document architectures, database models, high-level

communications and synchronization protocols, and real-time physical storage of multimedia data [Ben, thesis]. An interactive multimedia newspaper application has been implemented on the MEDIABASE platform. We have fit the broadband multimedia TeleLearning system into the MEDIABASE project so that implementations on lower-layer communication, as well as the database schema can be made use of.

### 5.1.1 The MEDIABASE System

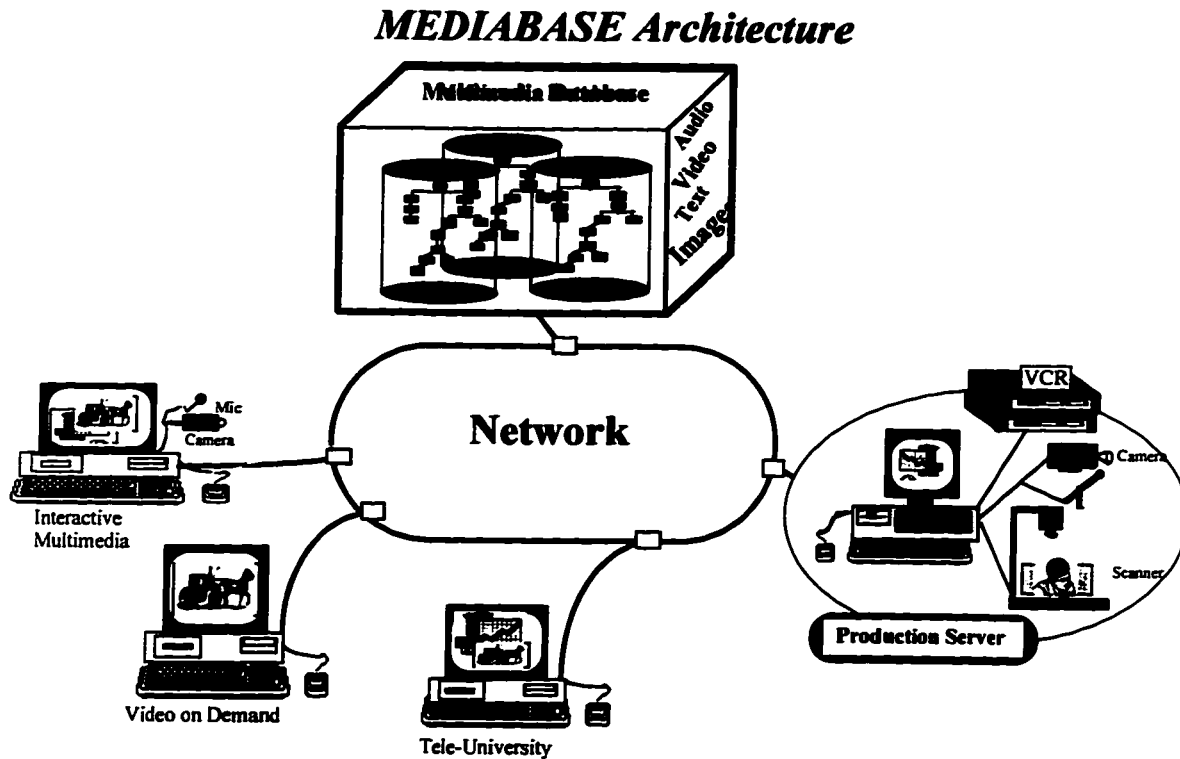


Figure 5.1 A generic architecture of the MEDIABASE system

In this section, the existing system components of the MEDIABASE project are to be described. However, our purpose here is only to give a brief description of the system on which our implementation mostly depends. Further details should be explored through

references we give in the context. Figure 5.1 shows the generic architecture of the MEDIABASE system. The main components include the following:

### **MEDIADOC**

A powerful multimedia document model capable of structuring information as well as representing the spatial-temporal relationships that may exist between different media objects. The model is designed based on the ODA logical structure and layout structure, with an extended time-line graph for the specification of the rendering scenario [James, thesis]. Examples of multimedia documents are stories, courseware, video clips, textbooks, electronic news, movies, medical reports, etc. [Ben, thesis].

### **Production Server**

A production server, as a part of the system, is responsible for capturing information from different resources and creating media objects as well as multimedia documents. Typically this component is equipped with advanced electronic devices as well as software modules to capture and display various media such as audio, video, animation, graphics, etc. The media objects as well as the documents produced by these modules may be stored locally or transferred over the network to the multimedia database for storage.

### **MEDIASTORE and MEDIAFILE**

These are multimedia database and file systems which provide the low level storage and database functionality of the system. It also handles synchronization, querying, and the real-time constraints. The specification of the meta-data (e.g., data

model, indexes), as well as the information content is required. The meta-data indicates the data structure that is used to store information in the database.

### **Multimedia Communication System**

For transferring multimedia documents over the network and handling the remote interaction between the user and the multimedia database, a multimedia communication system is developed. Appropriate bit rates for each type of media and synchronization are the key requirement. Furthermore, protocols which provide error-free and jitter-less transmission are desirable.

### **Multimedia User Interface**

The presentation of multimedia information to the users and the interaction between the users and the multimedia system is achieved through a multimedia user interface. Different multimedia application profiles may have various specific features that the user interface should support.

All these components are distributed over the OCRInet- an R&D ATM network in the Ottawa region, and a multimedia news application has already been developed.

## **5.1.2 Fitting a Broadband Multimedia Courseware Delivery System into MEDIABASE**

MEDIABASE system has already provided the document model, database structure, synchronization mechanism, as well as the network infrastructure for a broadband multimedia application. Therefore, the first prototype of our broadband

multimedia TeleLearning system is implemented on the MEDIABASE system. Meanwhile, we have also made some modifications and improvements in the infrastructure so that the resulted system is more fit for the requirements of a multimedia TeleLearning system. Figure 5.2 shows the architecture of the sub-system we have designed.

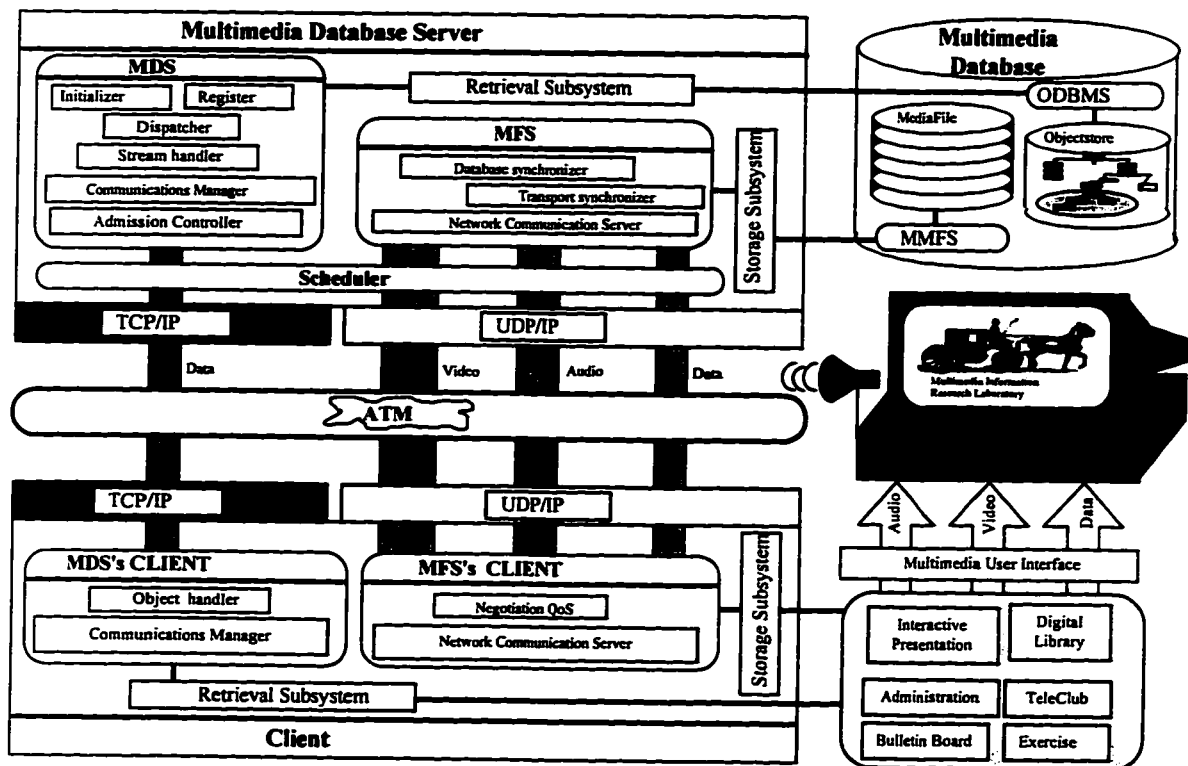


Figure 5.2 Interactive multimedia courseware platform

The implementation makes use of the ATM network and the communication protocols (TCP/IP/UDP) for communication, the commercial product ObjectStore (from Object Design) on a SUN/ULTRA workstation for the object-oriented schema to store the courseware, as well as a client-server model to access the database. All these components are already well defined and implemented in the MEDIABASE system. For a TeleLearning application, we consider that a student will access the virtual school from a

Personal Computer in most cases. Therefore, the courseware navigator was implemented on a multimedia PC with Windows 95. A new client module is implemented on PC for accessing the courseware database. The implementation of a courseware delivery system requires the following specific tasks to be fulfilled:

A courseware navigator providing a user-friendly graphical user interface for the learning environment.

A client module, which is embedded in the navigator program at the courseware user site, to provide APIs for accessing the database.

A courseware database which stores the course material as well as the students' information for administration.

## **5.2 Design Issues of a Courseware Navigator on Windows 95**

To access broadband applications from PC is a new trend which has enhanced the practicability of such applications. The reason lies mostly on the recent enhancement in the multimedia processing capability of PCs, and the big difference in the cost of workstations and PCs. The goal of a courseware navigator on a PC is to provide through a user-friendly graphical interface all the interactive learning services for students to learn flexibly, actively, and individually with on-demand help available.

## **5.2.1 Feature Set Analysis and Design**

The features to be provided to the students in a courseware navigator are analyzed. We have chosen some real-world education terms such as “classroom” and “library” to make the environment familiar with students. Meanwhile, new concepts discussed in this thesis are added to the implementation of those modules. The features provided can be categorized as follows.

### **Administration**

All the students should register at the virtual school before he can use the education facilities provided. In this way, it provides some useful data for the administration of the system and the school. On the other hand, this leaves some space for the further studying and development of the billing services for the TeleLearning applications. Each time a student accesses a course, it is required that the student number which identifies his registration should be provided. A student can update the information on his course registration, as well as personal data such as mailing address. Since all the student personal information as well as the courses they have registered for is stored in the center database, some statistics about the school, the course and the students themselves should also be available upon the students demand.

### **Classroom Presentation**

A player or a browser is provided to present the interactive multimedia courseware authored by the educators. This is the main place where the students acquire knowledge. This module should be able to handle the playback of different types of

media objects, while support the interpretation of links and actions between media objects and users' input.

### **Library Browsing**

Textbooks, reference books, and other related documents in any kinds of media types should be provided for the students to browse. By browsing the documents in the library, students can get extended information on topics which they are interested or related to the course content. Sometimes, new areas of interests may be found and explored provided with the strong cross-reference capability of the hypermedia information structure.

### **Meeting and Discussing**

The meeting and discussing module provides an environment for the students and the on-line consultants to communicate with each other. The students can use this facility to ask questions to the on-line consultants, or discuss or exchange their ideas with other students on a commonly interested topic. E-mail, telephone, and multimedia conferencing facilities are provided for the students to choose from according to the resources available on their platforms.

### **Bulletin Board**

When information is to be published to all the students, bulletin board should be used. For example, announcement of new courses or features of the virtual school, analysis of the common mistakes in an exercise, etc. We use news group to achieve this feature.

### **Exercise**

Practicing is the best way to learn. Some exercises can be authored in the courseware presentation itself. For example, courses presented using simulation-based teaching architecture such as drivers training course. However, for courses like mathematics which need practice the more the better, exercises can be provided as a separate module. Problems designed for the exercises can be in various styles besides the traditional text-based one. Contest can also be organized to stimulate the interests of the students.

### **Other Features**

In addition to the main features described above, some other features can be also provided for the convenience of the students' navigation. Bookmarks, which save the location of the interesting topics or media objects found during browsing, can be used. Mechanism to keep record of the stop position of the course presentation is also a possible service to be provided. With this facility, the courseware can automatically start the course presentation at the right place when a student enters again.

## **5.2.2 Multimedia in Windows 95**

It is very important for the courseware navigator to be able to display media objects in different types. Windows 95 provides a standard architecture for the creation as well as playback of digital video, digital audio, and MIDI which allows for high-quality multimedia effects. Usually, multimedia files are maintained in one of the formats described in table 5.1.

Format	Corresponding filename extension
Digital-video	.AVI
Waveform-audio	.WAV
Musical Instrument Digital Interface (MIDI)	.MID

Table 5.1 Multimedia files and formats

A WAV file is a Microsoft standard file format generally used for recording non-musical sounds, such as the human voice or a car horn [Charles, 95]. It can store about 1 second of sound in 11KB of disk space, or one minute of sound in 1MB of disk space. As the result, this medium tends to be extremely disk-intensive and is used mostly for adding short sound effects to a program or to the entire Windows environment. MIDI stands for Musical Instrument Digital Interface. A MIDI file can store one minute of fairly high quality musical sound in about 5KB of disk space which is only about one-twentieth space that of the WAV file. Therefore, MIDI files are much more useful than the WAV files for many purposes. The standard format for digital video in Windows 95 is AVI which stands for Audio Video Interleaved. AVI files can be used to store a motion video stream and one or more audio channels. Windows recognizes video data at varying resolutions, color depth, and refresh rates.

For developing applications to handle multimedia editing or playback, a Media Control Interface (MCI) is provided. MCI provides a device-independent command-message and command-string interface for the playback and recording of audio and visual data. Command-messages and command-strings can be used interchangeably. Commands such as stop, play, or pause are used to manipulate the media objects in different file formats.

With OLE (Object Linking and Embedding) functionality, sharing of data between OLE-compliant applications is enabled. An OLE document can be created to contain multiple types of data from different sources and several applications. Displaying or editing of those data is also allowed without running other applications. A multimedia document may include media objects in different types and coded with different formats such as MPEG for video, Waveform for audio. As long as the OLE-compliant player application of that object type is available on Windows 95, this media object can be played back by its own player when necessary.

### **5.3 Application Development**

We have chosen Visual C++ to be the environment for developing the software modules of the navigator application on Windows 95. The advantages of Visual C++ is that it provides an interactive application development environment as well as many efficient ways- seamlessly integrated in the environment, to simplify object-oriented programming and provide reusing code and components. Major facilities provided by Visual C++ includes the following:

- **Microsoft Developer Studio:** This is an interactive development environment. It has new class navigation tools- such as Class View and WizardBar to visualize the object class hierarchy as well as member attributes and member functions of each class. Also, AppWizard is provided for a fast start to build the application.

- **Component Gallery:** It is a one-stop location to store and to manage the reusable components, including C++ classes, OLE controls, and dozens of third-party components that enable you to quickly add advanced features.
- **Microsoft Foundation Class (MFC) library version 4.0:** MFC library contains more than 150 classes and 120,000 lines of code which is ready to be used. The library includes classes for database, communication, graphics, data model, file services, as well as documents and views. It also supports for creating applications that take advantage of the interface for Windows 95.
- **Database and cross-platform support:** Visual C++ includes Data Access Objects (DAO) classes for fast object-oriented access to the Jet database engine, and also ODBC support for creating database-independent, client-server applications. Cross-platform support is also available for Macintosh and other platforms.

In one word, Visual C++ can provide visual, reusable, therefore efficient application development.

We have started the development from using AppWizard to automatically construct the framework of the application. We have chosen Single Document Interface (SDI), with OLE support, and no database support for the framework. The next step is to construct a GUI as well as the client module to access the database. Moreover, special classes are designed and constructed according to the application requirements.

### **5.3.1 User Interface Construction**

The service that the navigator has to provide is to obtain the messages from the users input, then access the database through a client-server model to execute specific tasks, and finally output the result to the users. Therefore, most of the work done for the courseware navigator developed on Windows 95 is the construction of a user-friendly GUI. Most of the interactions for communicating between users and the application are provided through dialog boxes. Included in a dialog box, various controls can be used for different styles of interaction. For example, an animation control can be used for displaying video clips coded in AVI format; a button control can be used to react with users clicking; a list box control can list all the courses provided by the school, and let users to select one by one the courses he would like to register for. For displaying the courseware data retrieved from the database, we use the Internet Explorer. This is only the style for displaying HTML document. For other media objects, appropriate applications available on Windows 95 are to be activated.

### **5.3.2 Client Module Construction**

A client module is implemented for accessing the courseware data in the database on a SUN/ULTRA station. Details of the communication protocols are not going to be addressed here. Two APIs are provided by the client module for the applications built on top of its services.

- *Get\_List\_Doc()* is a function for getting the list of documents which are stored in the database. The function specifies the parameters for the message first, and then call the

*client()* function to communicate with the server through the network. The received message includes data which is the list of documents in which each name of a document is separated by a '\0' character. The function finally displays the list of document in a listbox.

- *Get\_Selected\_Doc()* is a function which can get a document from the server according to a specific document name, and then store the document in a temporary file locally in the user environment. The function also starts from specifying parameters for the message to be transmitted to the server, then calls *Client()* function to access the database, finally interprets the data received properly and saves it into a document. The only type of media supported now is HTML document.

More APIs are required to support more complicated functions for the navigator to access the database. However, these are left for the future work.

### **5.3.3 New Application Classes Design**

For the TeleLearning application, two special object classes are designed- CStudent class and CCourse class.

The CStudent class is designed for keep record of all data about a registered student which includes information about the student himself as well as courses he has registered. The information is initially given by the students in registration process, and can be updated later on. Although data is updated at the PC side through the interface of the navigator, it is also modified at the database side immediately through the client-server module. Some member functions supported by the CStudent class are: Constructor

for a new student, Constructor for registered student, Destructor, OnRegister( ) which provides the process for students' registration, and FindNumberOfCourse( ) for getting the number of courses the student has registered for.

The CCourse class is designed to keep record of courses a student has registered for. Course name, planned session to finish a course, course code, as well as the program which provides the courses are member variables in the class.

## **5. 4 Sample Learning Sessions**

In this section, a sample learning session is described in which a student will experience through the navigator application in our distributed multimedia TeleLearning system.

A student starts the learning session by running a navigator application installed on his computer. In our environment, they double click navigator.exe or a shortcut icon stands for it. A dialog shown in Figure 5.3 will be displayed on the screen. A video clip called "welcome" with synchronized music will be displayed in the animation window. The student need to type in his student number to access the virtual school, while a new student who has no record in the virtual school will have to register first. Other buttons are also provided for the students to watch a video clip of the general introduction of the virtual school, or version information about the navigator software.

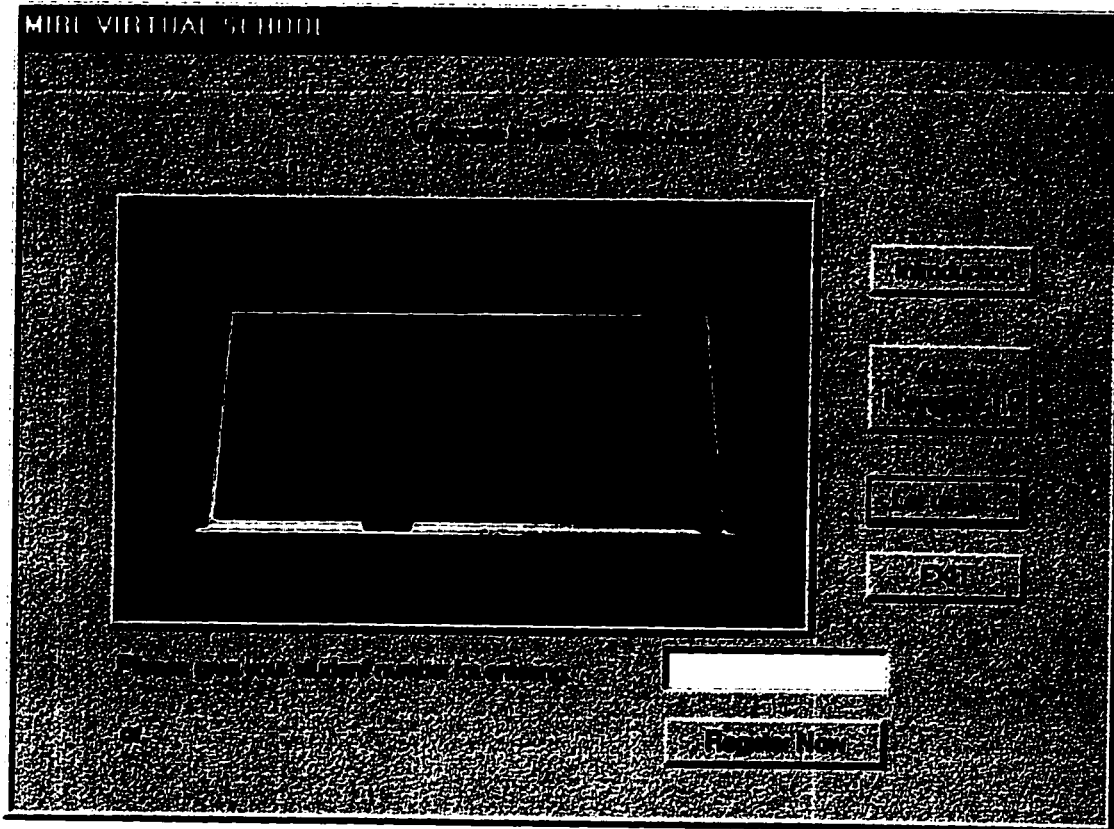
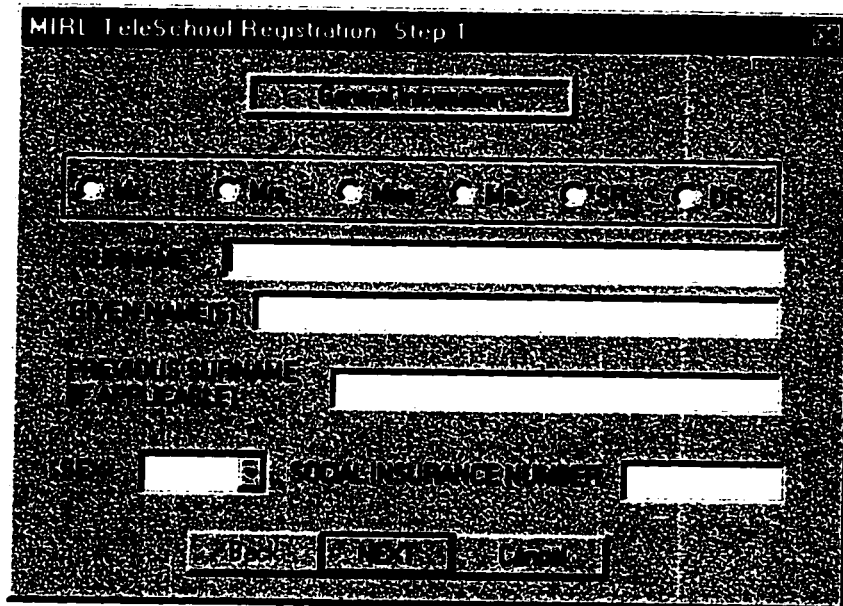


Figure 5.3 The first screen of the navigator

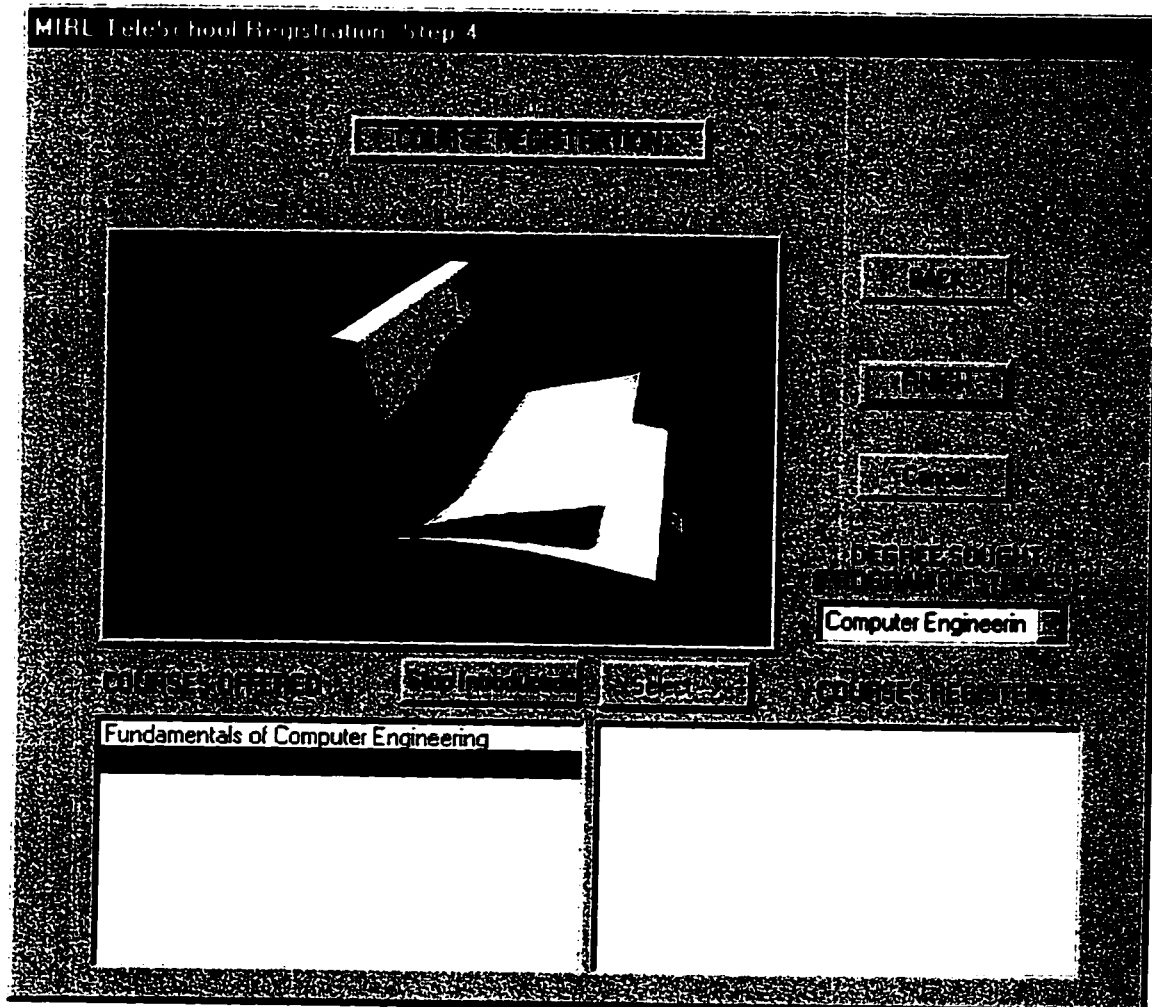


(a) Registration of general information

Figure 5.4 MIRC TeleSchool registration screen



In course registration dialog (Figure 5.4d), the student can choose a program, and get a list of courses provided in that program. Course introductions are created in multimedia formats and stored in the database. By selecting a course, then clicking the “introduction” button, a video clip is going to be shown to introduce the course objectives, content, as well as the teaching strategies. When the student decides to register for a specific course, he can use the “select” button to fulfill the task.



(d) Registration for courses

Figure 5.4 MIRL TeleSchool registration screen

Having finished the registration, the student is given a new student number and a choice for continuing or quitting the learning session for the moment.

Once the student has entered the TeleSchool successfully, all the facilities, including administration, classroom presentation, digital library, on-line help, can be accessed by the student through the main window. The student can change his/her profile through the interface provided in "administration" (Figure 5.6). They can also access the courseware presentation (Figure 5.5) as well as browse the library (Figure 5.7).

After finishing a learning session, the student can click "exit" to terminate the program. Some important information such as the stop position of the courseware presentation is to be automatically stored for later usage.

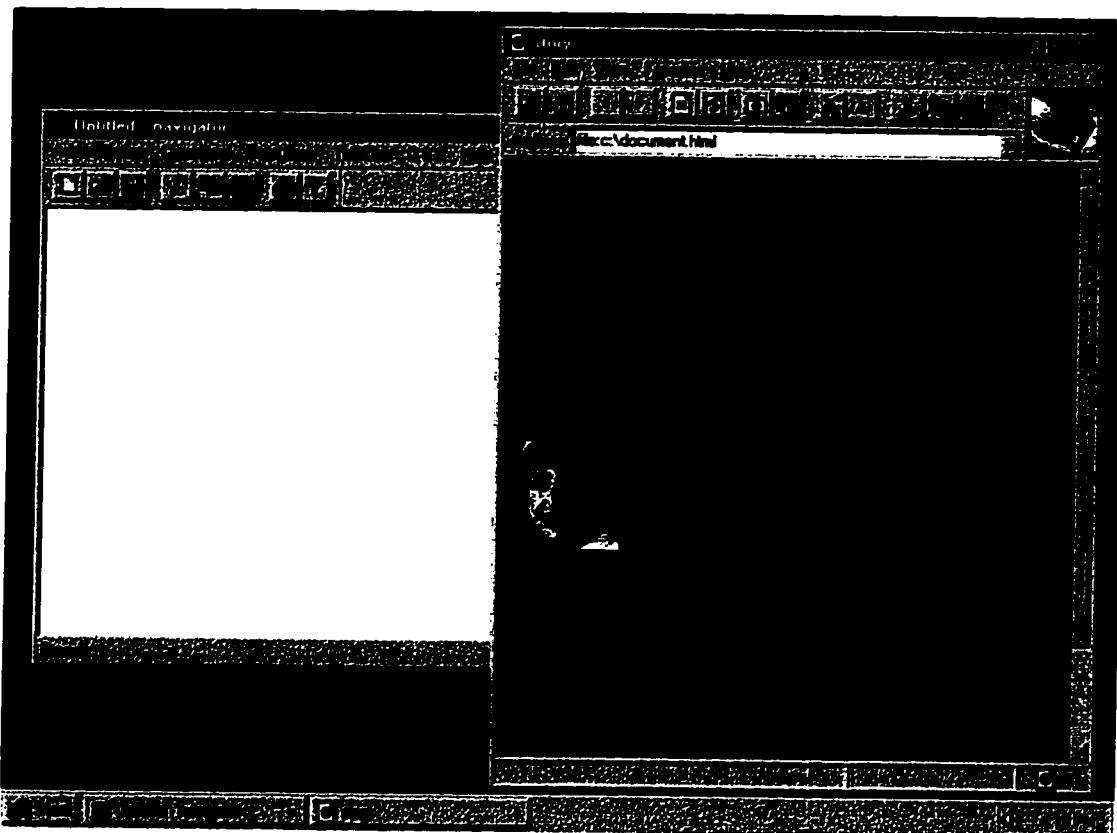


Figure 5.5 The course presentation screen

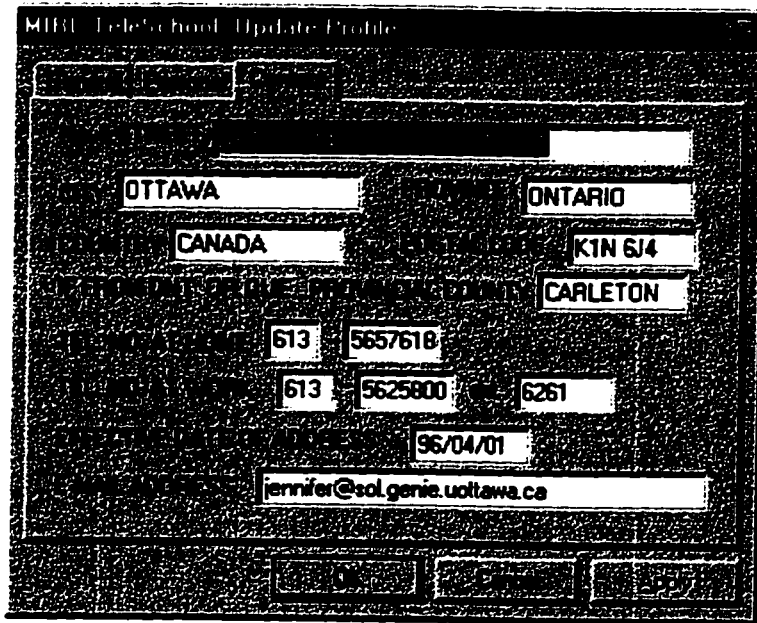


Figure 5.6 The screen for updating the student profile

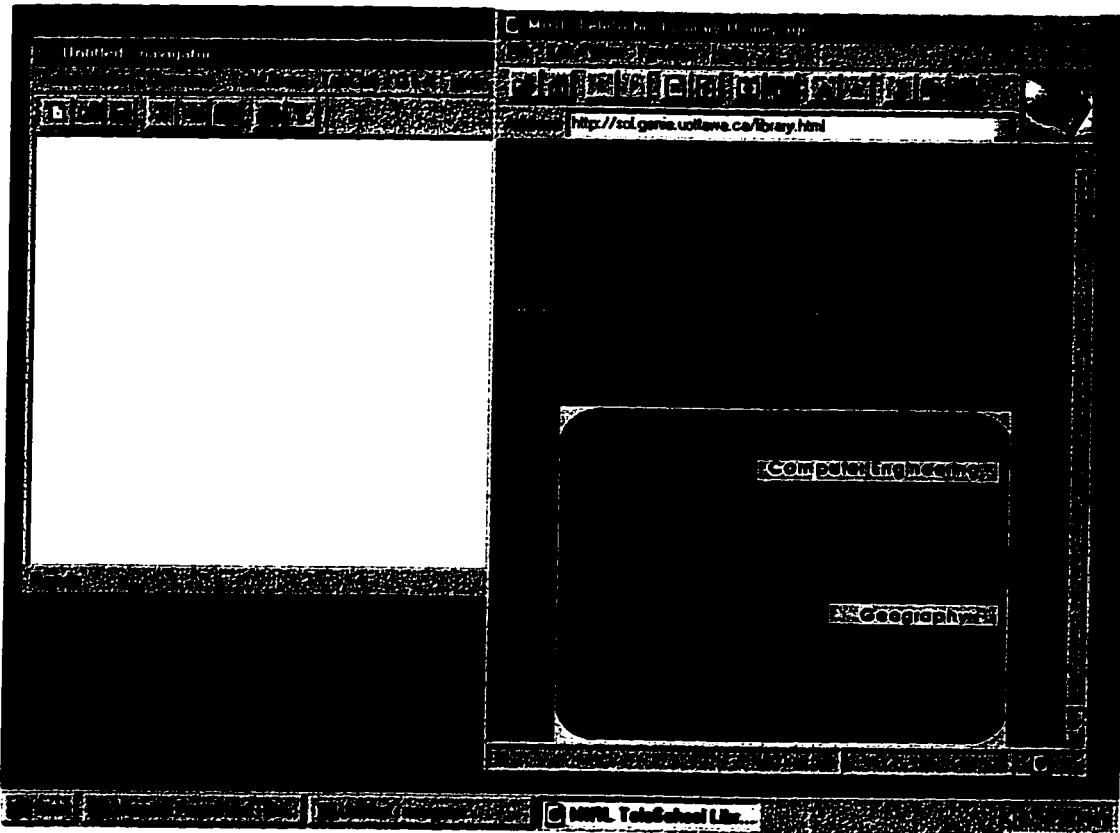


Figure 5.7 The screen for browsing the library

## **5.5 Summary and Related Work**

In this chapter, design and implementation of a courseware navigator in a broadband multimedia TeleLearning system is introduced. We have studied the capability of Windows 95 in processing multimedia information, and some experiences are obtained from the implementation. However, the implementation finished by now focuses more on GUI design and the framework of the whole program. Further studies and practices should be made on multimedia data handling and interpreting, for example, how to get video clips from the database and then play it back. Also, along with the construction and the development of the database, the client module should also be enhanced. More APIs are to be provided, for example, query functions such as `GetKeywordTree( )` to retrieve and display the keywords provided by the database; `GetDocByKeyword(CString Keyword)` to get the document list in the database by the keyword provided, etc.

## Chapter 6

### Conclusion

#### 6.1 Summary

The world has been changing fast during the past few decades. In leading most of the changes, information technology is reshaping the way people live. The fast explosion of the use of Internet is one of the evidence. Currently, people are talking about TeleEducation, TeleShopping, TeleCommuting, etc. We are excitedly expecting a new world in which a computer at home or in the office can provide access to a remote system and do all the work, while in a more convenient and lively way. However, blindly move the currently existing real-life systems into computers does not mean that we can have a better environment for information processing. What makes computer applications superior to the existing systems is its non-linear knowledge processing capability, multimedia information processing capability, as well as interactivity provided in

applications. These are the basis of the Multimedia Interactive TeleLearning System we have designed in the thesis.

With the advancement of technologies, learning in the future can be radically different from what it is like today. The Multimedia Interactive TeleLearning System introduced in this thesis, as the integration of the modern learning theories and the advanced technology, has put much effort in making education and training more accessible, more flexible, more individual and as the result, more effective. The system is able to offer distributed, multimedia and interactive training and education services. Users can be freed from those constraints brought by time, location, or platform which are common in most of the emerging computer applications for education. Moreover, it also provides more facilities for the education organizers to manage the courseware, trace the students' progress, and offer facilitation when necessary.

The MHEG standard, designed for use and interchange of real-time, reusable multimedia/hypermedia information by applications in a wide range of domains and on heterogeneous platforms, is an important standard addressing information processing of interactive applications. From the study undergone in this thesis, we have seen lots of advantages of MHEG in delivering real-time, interactive, multimedia information. Although, MHEG standard is still under development, and it might be too early to determine its future, we still consider it as a promising method for the future multimedia/hypermedia applications.

## 6.2 Suggestions for Future Work

TeleLearning is an inter-discipline area which involves education theory, real society requirements, telecommunication technology, as well as information processing technologies. This thesis work has only opened a window for perspecting the area. We have gained quit a lot of precious conclusions and experience. However, there are still lots of work to be done before a really better education environment, as we have aimed at all through our work, can be finally achieved.

In courseware authoring, we have introduced some teaching architectures proposed by Schank [Schank, 94]. Two interactive document models as well as object libraries are designed for the implementation of these teaching architectures. However, the mapping of concepts and implementation details from each layer to its next lower layer has not yet been studied. This is an important detail design part for the courseware editor which is to be provided in the TeleLearning environment.

The object libraries we have designed is based on the understanding of MHEG standard part I, as well as the courseware authoring requirements. Since other part of the MHEG standard was not available at the time this work was done, we were not able to be very precise and complete for the work. For example, script object class was not studied because of the unavailability of materials and standards. On availability of other parts of the MHEG standard, with the further analysis and development of the courseware authoring system, new object classes are to be added. Furthermore, adjustment of the class hierarchy may also be necessary.

Apart from what we have discussed above, additional features can be integrated into the system based on the open system infrastructure we have designed. *Multimedia collaborative document editing* can be used by both courseware authors and students for joint authoring of an interactive multimedia document. Some features such as *multimedia library, exercise and feedback facilities, and multimedia conferencing*, also need further study and development although feature description has been finished in this thesis.

## Bibliography

- [Apple, 79] Apple, M., *Ideology and Curriculum*. Boston: Routledge & Kegan Paul. (From [Schank, 90])
- [Schank, 90] Roger C. Schank, Menachem Y. Jona, *Empowering the Student: New Perspectives on the Design of Teaching Systems*. Northwestern University, Institute for the Learning Science, Technical Report #4, August, 1990
- [Grace, 95] Grace Au, *Can Multimedia Help People Learn Faster?* Proceedings of the IEEE international conference on multimedia computing and systems, May 1995.
- [Schank, 94] Roger C. Schank, Active learning through multimedia. *IEEE Multimedia*, spring 1994, pp.69-78
- [Landow, 90] Landow, G.P., *Hypertext and Collaborative Work: The Example of Intermedia*. In Galegher, J., Egidio, C. And Kraut, R. (1990), *Interlectual Teamwork*. Lawrence Erlbaum, 1990, Hillsdale, NJ.
- [Marchionini, 94] Marchionini, G., *Evaluating Hypermedia and Learning: Methods and Results from the Persus Project*. *ACM Transactions on Information Systems*, 12, 1, (1994), 5-34

## Bibliography

---

- [James, thesis] James W. Emery, *Design and Implementation of a Time-based Multimedia Document Architecture*, 1994
- [Multitorg, 95] Dag Solvoll, Geir Ivarsoy, Hakon W. Lie, and Per E. Dybvik, *Information Exchange in MultiTorg*, [http://www.nta.no/elektronikk/4.93.dir/solvoll\\_D.html](http://www.nta.no/elektronikk/4.93.dir/solvoll_D.html)
- [Greg, 93] Luft Greg, *Multimedia in Education*, First Light Video Publishing (Audio-Visual), 1993
- [HyTime, 91] ISO/IEC CD 10744, "*Hypermedia/Time-based Structuring Language (HyTime)*" (Committee Draft), April 1991.
- [John, 93] John F. Koegel, Lloyd W. Rutledge, John L. Rutledge, Can Keskin, *HyOctane: A HyTime Engine for an MMIS*, ACM Multimedia 93, pp129-136
- [Steven, 91] Steven R. Newcomb, Neill A. Kipp, and Victoria T. Newcomb, *The "HyTime": Hypermedia /Time-based Document Structuring Language*, Communications of the ACM, November 1991 vol.34 No. 11, pp67-83
- [Brian, 92] Brian D. Markey, *HyTime and MHEG*, Digest of Papers- IEEE Computer Society International Conference 1992, pp25-40
- [Avner, 79] A. Avner, *Production of Computer-based Instructional Materials*, Issues in Instructional Systems Development, H.F.O'Neil, Jr. (Ed.), pp. 133-180, Academic Press, New York, 1979 (from [Rada, 96])
- [Rada, 96] Roy Rada, *System for Reusing Hypermedia Courseware*, Multimedia Tools and Applications, January 1996 v2 No. 1, pp53-78
- [MHEG, 95] ISO/IEC DIS 13522-1: 1994, *Information Technology- Coding of Multimedia and Hypermedia Information Part 1: MHEG Object Representation, Base Notation (ASN.1)*, 1995

## Bibliography

---

- [Roger, 93] Roger Price. *MHEG: An Introduction to the Future International Standard for Hypermedia Object Interchange*. ACM Multimedia 93, pp121-128
- [Nael, 95] Nael Hirzalla, Ben Falchuk, and Ahmed Karmouch, *A Temporal Model for Interactive Multimedia Scenarios*, IEEE Multimedia, Fall 1995, pp24-31.
- [Ben, thesis] Benjamin W. Falchuk, *MULTINoDes- A Multimedia News-On-Demand System*, 1995
- [Smith, 81] Smith, K., Johnson, D.W., & Johnson, R.T., *Can Conflict Be Constructive? Controversy Versus Concurrence Seeking in Learning Groups*. Journal of Educational Psychology, 73, 651-663. [Schank, 90]
- [Citizen, 96] *University Without Walls*, Ottawa Citizen June 9, 1996
- [Francoise, 94] Francoise Colaitis, *Opening Up Multimedia Object Exchange with MHEG*, IEEE Multimedia, Summer 94, pp.80-84
- [Thomas, 95] Thomas Meyer-Boudnik, Wolfgang Effelsberg, *MHEG Explained*, IEEE Multimedia, Spring 1995, pp26-38.
- [Ben, 95] Ben Falchuk, A. Karmouch, *A Multimedia News Delivery System Over an ATM Network*, Proceedings of the IEEE international conference on multimedia computing and systems, May 1995.
- [Midoro, 93] Midoro V., *What Makes Hypermedia Systems Interesting for Education?*, Educational Multimedia and Hypermedia Annual, 1993
- [Dexter, 94] F. Halasz and M. Schwartz, *The Dexter Hypertext Reference Model*, Communications of the ACM, pp. 30-39, ACM Press, New York, 1994
- [Joekyne, 95] Joekyne Nanard, Marc Nanard, *Hypertext Design Environments and the Hypertext Design Process*, Communications of the ACM, August 1995 v38 No. 8, pp49-56

## Bibliography

---

[Petra, 92] Petra Hoepner, *Synchronizing the Presentation of Multimedia Objects*, Computer Communications, November 1992, vol 15 no 9

[Charles, 95] Charles Calvert, *Teach Yourself Windows 95 Programming in 21 Days*. SAM Publishing, 1995

## Publication

1. R. Wang, A. Karmouch, *A Broadband Multimedia Telelearning System*, Proceedings of 5th Int. IEEE Symposium on High-Performance Distributed Computing- Multimedia and Collaborative Environments, Syracuse, U.S. A., August 7-9, 1996
2. R. Wang, A. Karmouch, *A Broadband Multimedia TeleLearning Platform*, Proceedings of the 18th Biennial Symposium on Communications, Kingston, Canada, June 3-4, 1996
3. R. Wang, A. Karmouch, *Multimedia Courseware Delivery Over Broadband Networks*, Proceedings of the IEEE ICC '97, Montreal, Canada, June 8-12, 1997.