

## DESIGN AND IMPLEMENTATION OF IMPORTANT APPLICATIONS IN A JAVA-BASED MULTIMEDIA DIGITAL CLASSROOM<sup>1</sup>

Kuang-Chih Lee, Keh-Ning Chang, Shiu-Sheng Yu,  
Ing-Chau Chang, Chee-Wen Shia, Wen-Chin Chen, and Jau-Hsiung Huang  
Communications and Multimedia Lab.  
Department of Computer Science and Information Engineering  
National Taiwan University, Taipei, Taiwan

### Abstract

This paper presents important issues on the design and implementation of a Multimedia Digital Classroom (MDC) system with its applications. The MDC system is implemented in Java language. It aims at providing important tools to support a distributed interactive education environment. The system consists of a classroom manager and some powerful teaching application tools. The classroom manager is a sharing system with a database. The teaching application tools include of Multimedia Internet Browser, Shared White Board, Multi-Channel Audio Conference, WWW-Based Note-Taking System, and Instruction Evaluation System. The MDC system and these application tools have been implemented and tested on the heterogeneous environment where workstations and PCs are connected through networks.

### I. Introduction

As the rapid development of Internet and computer systems, there are more and more multimedia applications in markets. Some of these applications help geographically dispersed people working for their common goals. The distance learning application, which is one of them, plays an important role of education popularization. It has been predicted that the distance learning application would be one of the most important commercial applications in next decade [1]. It breaks the limitation of traditional courses so that teachers and students do not need to be in the same classroom. These geographically dispersed teachers and students in the digital classroom can learn and share knowledge with less effort and cost.

In the digital classroom, a variety of functionalities should be provided to support all types of learning processes, for example, lectures, discussions, examinations, and so on [2]. With the unique characteristics of platform

transparency and application protocol integration, World Wide Web (WWW) technologies have been adopted for information sharing on the cyberspace. Hence, this paper presents important issues on the design and implementation of a WWW multimedia digital classroom and its applications using Java language. The multimedia digital classroom includes many teaching application tools, such as Multimedia Internet Browser, Shared White Board, Multi-Channel Audio Conference, WWW-Based Note-Taking System, and Instruction Evaluation System.

Organization of this paper is as follows. In section two, we will first propose system architecture of the MDC system; then designs of teaching application tools are illustrated there. In section three, implementation issues of the MDC system and applications are described. Further, difficulties met in implementation and how we trade off between these difficulties are discussed in section four. Finally, we conclude this paper in section five.

### II. Design of the MDC System and Its Applications

System architecture of our Multimedia Digital Classroom (MDC) [3] is shown in Figure 1. There is one teacher and numbers of students in a classroom. The teacher first creates a classroom on a host. Whenever a student wants to join the classroom, he/she has to register with the teacher's classroom server to create a connection with the teacher. There are two major components in Multimedia Digital Classroom. One is the classroom manager, and the other is a package that is composed of many teaching application tools.

#### A. Classroom Manager

The classroom manager is responsible for construction of classroom environment and interception, transmission,

<sup>1</sup> Research supported in part by National Science Council of R.O.C. under grant No: NSC85-2662-E002-015

reconstruction of events from teaching application tools. It is a sharing system with a database. The sharing system uses the “*event-sharing*” model and provides a framework so that a single-user application can be executed cooperatively in a heterogeneous environment. The event-sharing model presumes that the same application is started locally by all session participants. For achieving the same views of these applications, current token holder’s inputs to these applications are first intercepted by the local shared manager and transmitted to the classroom server of the teacher. Then the classroom server broadcasts these input events to the shared managers of all students, through connections among the teacher and students. These input events are executed by their associated application in each student’s computer to generate the same view of the application.

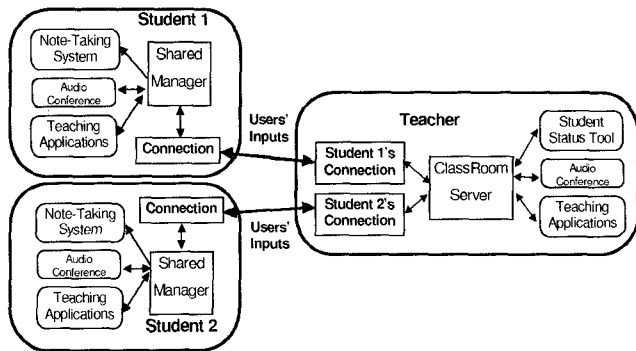


Figure 1. System Architecture of the Multimedia Digital Classroom.

Figure 2 shows a detailed diagram of the relation between the teacher and a student participant in event-sharing model. Originally, users’ mouse or keyboard inputs are directly handled by the application through Java Abstract Window Toolkit (AWT) class library. For sharing the application, these inputs are intercepted by the shared manager in the event sharing model. With this event sharing approach, less network bandwidth is needed for users’ inputs. This approach is especially suitable for the low network bandwidth environment, such as communication through modems and the telephone network.

In addition to the event sharing model, a hybrid sharing *framework* is proposed to enhance the event-sharing model and then let our classroom manager be more powerful and flexible. The framework provides three kinds of interfaces. The default interface adopts the event-sharing model. Applications using it do not need to modify. Another interface is called as “*PrivateCom*” (private component) interface. If any application implements the PrivateCom interface, its input events will not be captured by the

sharing system. It is useful for the work that is operated by the local participant only. For example, the examination system.

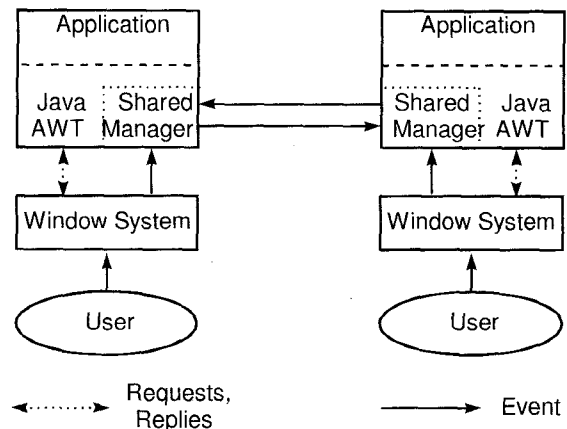


Figure 2. the Event Sharing Model

The other interface is called as “*ShareAction*” (share action) interface which defines one method, i.e., *doAction()*, to handle application specific actions. These actions are represented for complex operations, such as the URL “backward” operation issued in the web browser. These operations are encoded by the local application, transmitted through the network, and decoded by the remote application for execution using the *doAction()* method. Details of the *doAction()* method and the encoding/decoding rules are described in [4]. The method uses network channels to broadcast and synchronize actions to be shared. When all participants of classroom managers receive a *ShareAction* command, they call the *doAction()* method of the target component to manipulate the received command. The “*ShareAction*” mechanism can greatly reduce the complicated event processing of some components when the event sharing method is used. In this way, synchronization with remote applications could be more easily achieved with the “*ShareAction*” method [4].

As we mentioned before, the classroom server has a database. This database records course names, teacher names, student names, registration numbers, and other information of the class. Teacher or students can query class information stored in the classroom database.

### B. Teaching Application tools

All teaching application tools are started by the classroom manager. The sharing capability, which comes from the hybrid-sharing mechanisms of the classroom manager, is transparent to applications, such as Multimedia Internet Browser and Shared White Board.

We briefly describe each application as follows:

● **Multimedia Internet Browser**

The World-Wide Web (WWW) is tremendous popular recently. Multimedia Internet Browser provides a bridge between digital classroom and WWW. A teacher uses this web browser to open a hypertext document which contains the contents of the current course, i.e., the outline page, in the digital classroom. Each item in the outline page further links to other web pages, which with teaching materials included. The teacher can lead all students to study on the WWW by sharing the same view of the navigated web pages. Besides, the documents of teaching materials will be recorded by the note-taking system. The note-taking system will be described later.

Figure 3 shows a detailed diagram of the architecture of the Multimedia Internet Browser. Except for the parsing capability of the HTML format, it also provides the ability to browse VRML documents and play some kind of multimedia data, such as the MPEG-1 audio and video.

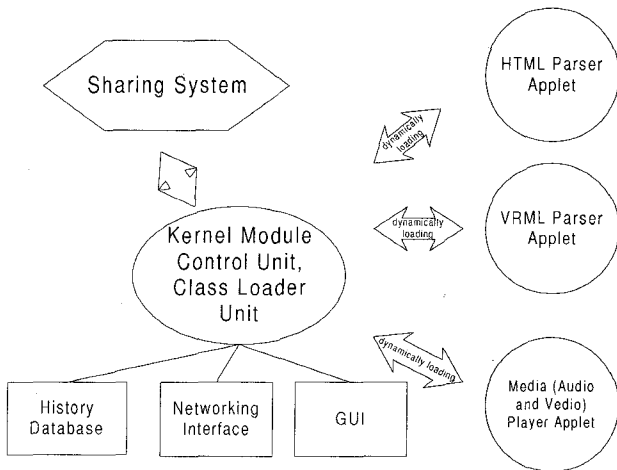


Figure 3. Architecture of Multimedia Internet Browser

● **WWW-Based Note-Taking System**

The note-taking system preserves important teaching materials easily. A scenario of taking notes in the MDC system is as follows. After the classroom initialization, each student can invoke the note-taking system to make his/her own note for this class. First, the teacher uses the browser to open the WWW outline page prepared in advance for teaching. Each item in the outline page links to the page that contains teaching materials. The teacher advances the teaching process by navigating pages through links of the outline and linked pages using the browser, with audio conference and other teaching application tools. The note-taking system first converts contents of the outline page to its internal format and helps the student to add his/her annotations for any item of the outline [5].

Whenever the teacher clicks an item in the outline page to open a new page, students' note-taking systems start recording audio data of audio conference, his or her own text annotations, timing information and all users' inputs to the teaching application tools into a file. The recording process for this item continues until the teacher clicks another item in the outline page. As the class course finally terminates, the note-taking system converts the generated note back to a WWW note page with each item linked to its corresponding file. Later, the student can use the browser to review notes by clicking an item to invoke the player. The player reads the file, parses the recorded information, and invokes associated applications to replay by re-executing the original teaching process synchronously [6]. Scenario of the note-taking process is shown in Figure 4.

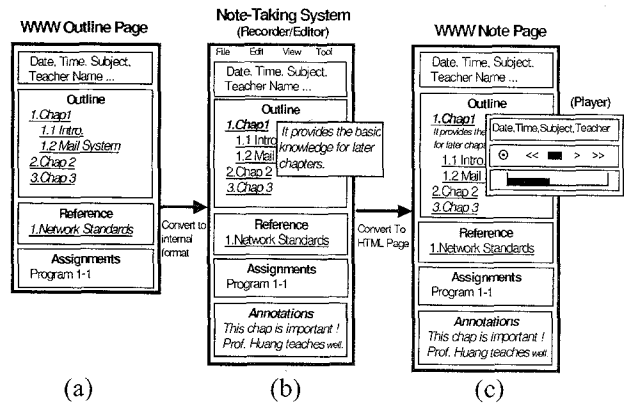


Figure 4. Scenario of the note-taking process. (a) Format of the outline page. (b) User interface of the note-taking system. (c) Format of the note page.

With this WWW-based design, notes can be easily shared with other students. Furthermore, recording operations are automatically handled by the note-taking system such that students can concentrate on the teaching materials with minimal distraction.

● **Multi-Channel Audio Conference**

Voice is a direct communication method traditionally. If teachers and students use voice to communicate with each other in the digital classroom, the learning process will be more efficient because students can ask some questions directly about difficult or unclear teaching materials.

The audio conference in our digital classroom is based on the client-server model. When the audio conference starts, voice uttered by all participants is recorded, compressed, and then sent to the audio server in the network. The audio conference server receives all audio source packets; then decompresses, mixes, and transmits

back to all participants. Finally, all users will hear the mixed voice by audio player device. Scenario of the audio conference process is shown in Figure 5.

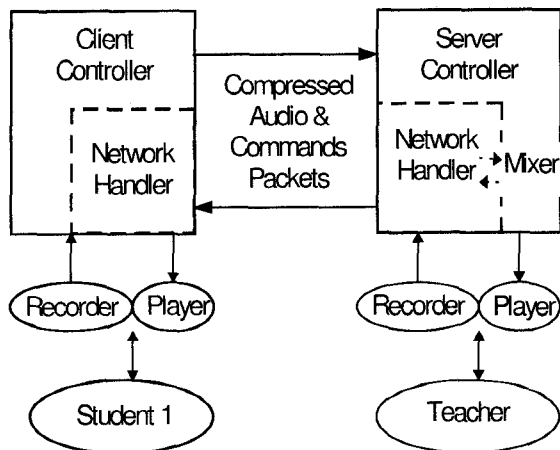


Figure 5. System architecture of audio conference process

● **Instruction Evaluation System**

Instruction evaluation, sometimes we called examination, usually takes place when teachers want to evaluate teaching utilization. In distance learning environment, examination remains important and needs to be given, and we must preserve the security of examination data in network transmission. Instruction Evaluation System consists of four modules. They are: PE (Proposition Editor), ASS (Answer Sheet Server), AE (Auto Evaluator), and AC (Answering Cookie).

The scenario of examining process in our digital classroom is as follows. First, a teacher uses our shared browser to open an examination paper of HTML format which is generated by PE; then all students see the same contents of examination paper and begin to answer. The layouts of each answer sheet are dynamically generated by the applet of AC. After the examination, AC extracts the answers from the answer sheet and encrypts them by DES encryption algorithm; then sends them to ASS. When ASS receives the encrypted answers, AE will decrypt, evaluate, and then calculate the scores of all students. Finally, the results of the examination are stored in the database of the MDC system. All students and teachers can query the database to get the score information. Figure 6 shows the process of the overall system.

It is important to preserve security of the examination system. With the DES encryption/decryption mechanism, we can prevent cribbing by intercepting the other answer in the network. Password checking avoids that other students use aliases to cheat in the examination.

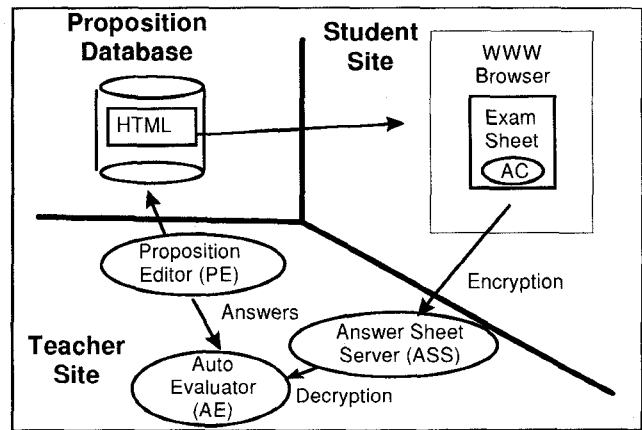


Figure 6. Architecture of Instruction Evaluation System

● **Shared White Board**

Because we have the support of sharing system environment, a simple drawing-panel application executed by the classroom manager becomes a shared white board without any change. Therefore, we implement a drawing panel with some basic functions, such as drawing basic graphics, writing texts, and multiple drawing pages. All people in the digital classroom using the shared white board at the same time do not interfere with each other.

**III. Implementation Issues of the MDC System and Its Applications**

The MDC system is implemented on a heterogeneous environment where workstations and PCs connected through an FDDI and an Ethernet network. We list the development environment as follows:

	PC	WORKSTATION
Operating System	Windows 95	Sun Solaris 2.5.1
Java Compiler	Symantec Cafe 1.51	SUN JDK 1.02
Native C Code Compiler	Microsoft Visual C++ 4.0	GNU C Compiler
Network	10 Base-T Ethernet	10 Base-2 Ethernet

We have tested MDC system with participants from Windows 95 with Ethernet, Windows 95 with modem and the telephone network, and SUN Solaris with Ethernet, respectively. The snapshot of MDC system is shown in Figure 7. The Shared White Board at the upper left-corner

and the Multimedia Internet Browser at the upper left corner are used for teaching. The Multimedia Internet Browser supports functions for the HTML 3.0 tag, VRML 1.0 tag, MPEG audio/video and several kinds of image formats. All modules of Multimedia Internet Browser described in section two are applet-based and dynamically loaded by the browser kernel. They are functionally independent between each other so that performance enhancement can be achieved by using the multi-thread technique to execute different modules in parallel. With applet-based module design, we can get the maximum extension for future improvement. New module, downloaded from web server as java applet, could be added into the browser dynamically.

User can also add other functions to the browser without changing anything if these functions are written by Java applet class.

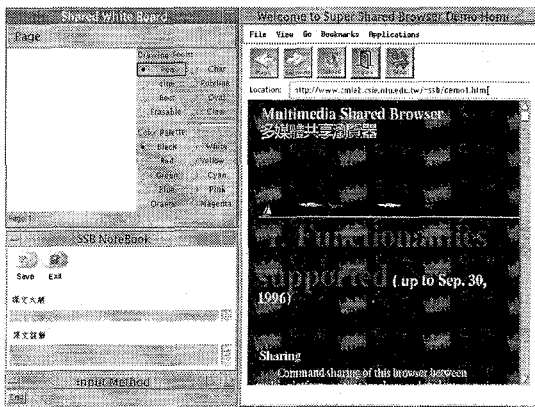


Figure 7. A snapshot of MDC system, including Multimedia Internet Browser, Note-Taking System, and Shared White Board.

The note-taking system, shown at the lower right corner in Figure 7, shares contents of the outline page with the Multimedia Browser. Students are allowed to type their own text annotations in Chinese or English in the lower window of the note-taking system. After the class, students can use the browser to review WWW note pages and click any item to invoke the player. The player is a Java applet which invokes associated applications to replay the recorded information synchronously. The interface of the player applet is shown in Figure 8.

When the teacher starts the Audio Conference, the control panel will appear, which is shown in Figure 9. The control panel shows how many participants are on this digital classroom, and who is speaking now. User can adjust the volume of recorder or player by dragging the controlling scrollbars. We use ITU-T G.723 [7] audio compression/decompression standard to reduce audio data

size. Compressed audio packets are transmitted by the UDP protocol. Besides, we use multiple threads for implementing the audio recorder, player, network handler, and central controller modules in order to enhance performance and avoid busy waiting situation. Moreover, a snapshot of the Instruction Evaluation System is shown in Figure 10.

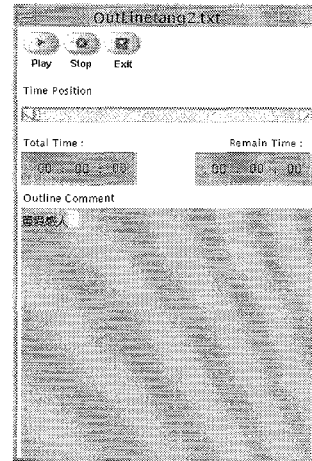


Figure 8. Interface of the Notes Player

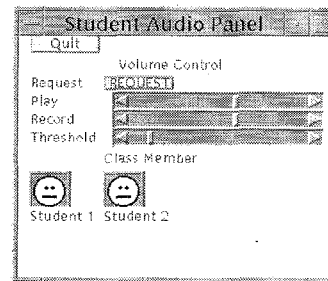


Figure 9. Control panel of the Audio Conference

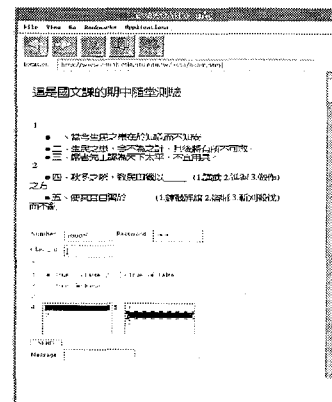


Figure 10. Snapshot of the Instruction Evaluation System

#### IV. Discussions

This section discusses important implementation problems and the approaches we took to resolve these issues.

##### ● Performance Challenge

In order to achieve platform transparency, Java adopts the *virtual machine* technology. Each Java supporting platform executes the virtual machine first. Then the virtual machine loads the class file and interprets these bytecodes [8]. Due to this kind of approach, performance of Java applications is usually slow in current environment. We use two methods to improve the performance. First, we override some Java AWT class libraries to reduce unnecessary repainting operations. Then, C native codes are used in CPU burst routines, such as compression and decompression processes of the audio data as mentioned before.

##### ● Weak Device Interface Supported by Java

Java lacks audio device interface and printer interface in JDK version 1.02. We use native codes to develop audio device driver, including the recorder and player. The method raises complexity of the overall system. Moreover, we must implement different device drivers in order to pertain the cross-platform property.

##### ● Network Limitation

The ideal network environment of the MDC system should have the following properties:

- Low delay latency
- Error free transmission
- Multi-points communication

However, Internet adopts the TCP/IP protocol to transmit data packets. TCP/IP protocols guarantee error-free transmission, but do not support bounded transmission delay and multi-points communication. Therefore, we use UDP to send the audio data that need real-time transmission. The MDC server constructs many point-to-point TCP connections for each client to transmit shared events and actions. Although this approach resolves multi-point communication problem, it wastes too much network bandwidth for transmitting duplicate packets on these point-to-point connections.

#### V. Conclusions

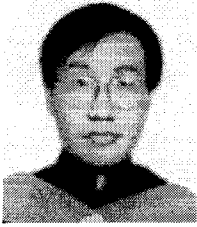
In the near future, distance learning will become popular in our daily life. The MDC system in this paper provides

miscellaneous applications to build a distributed interactive education environment. Consequently, we believe that the utilization of the MDC system will be powerful in the future. The future work in this MDC system is to integrate the video conference system, to provide linkage with school database, and reduce the complexity of this system by adopting advanced networking interfaces, for example, the ITU-T T.120 protocol suites [9].

#### VI. References

- [1] H. Meleis, "Toward the Information Network," IEEE Computer Mag., pp.59-67, Oct. 1996.
- [2] R.T. Kouzse, J.D. Myers, W.A. Wulf, "Collaboratories: Doing Science on the Internet," IEEE Computer Mag., pp.40-46, Aug. 1996.
- [3] Multimedia Digital Classroom Project in Communications and Multimedia Lab., Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, <URL:<http://cmlab.csie.ntu.edu.tw/~ssb>>, 1996.
- [4] Shih-Sheng Yu, Chee-Wen Shia, and Wen-Chin Chen, "Application Sharing Framework on Java," to be appeared in Proc. of IEEE International Conference on Consumer Electronics, Chicago, Illinois, USA, June, 1997.
- [5] R. Manohar and A. Prakash, "Replay by Re-execution : a Paradigm for Asynchronous Collaboration via Record and Replay of Interactive Multimedia Sessions", ACM SIGOIS Bulletin, Vol. 15, No. 2, pp. 32-34, Dec. 1994.
- [6] Ing-Chau Chang, Li-Li Chen, Ji-Jing Shen, Ke-Chau Hsu, Jau-Hsiung Huang, "Design and Implementation of a Multimedia WWW-Based Note-Taking System for Distance Learning", to be appeared in Proc. 1997 International Conference on Consumer Electronics, Chicago, Illinois, USA, June, 1997.
- [7] ITU-T Recommendation G.723, "Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s," <URL:[http://info.itu.ch/itudoc/itu-t/rec/g/g700-799/g723-1\\_32261.html](http://info.itu.ch/itudoc/itu-t/rec/g/g700-799/g723-1_32261.html)>, 1996.
- [8] Tim Lindholm, Frank Yellin, "The Java Virtual Machine Specification," Addison-Wesley, MA USA, 1996.
- [9] ITU-T Recommendation T.120 : "Data Protocols for Multimedia Conferencing," <URL:[http://info.itu.ch/itudoc/itut/rec/t/t120\\_35511.html](http://info.itu.ch/itudoc/itut/rec/t/t120_35511.html)>, July, 1996.

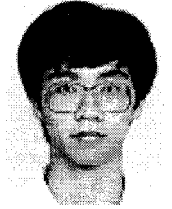
#### BIOGRAPHY



**Kuang-Chih Lee** received the B.S. and M.S. degree in Computer Science and Information Engineering from the National Taiwan University, Taipei, in 1995 and 1997, respectively. His research interests include multimedia systems, programming language, and Internet technologies.

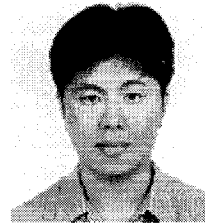


**Chee-Wen Shia** received the M.S. degree in Computer Science and Information Engineering from National Taiwan University, and is now studying for Ph.D. degree. His research interests include CSCW and multimedia systems.



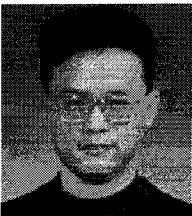
**Keh-Ning Chang** received the B.S. and M.S. degree in Computer Science and Information Engineering from the National Taiwan University, Taipei, in 1988 and 1992, respectively. He is currently working towards the Ph.D. degree in Computer Science and Information Engineering at National

Taiwan University. His research interests include algorithm and programming language.



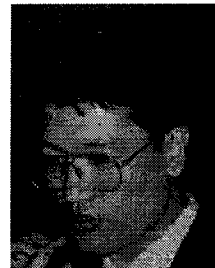
**Wen-Chin Chen** received the Ph.D. degree in Computer Science from Brown University. He is currently a Professor in the department of Computer Science and Information Engineering, National Taiwan University. His research interests

include algorithms and multimedia information systems.



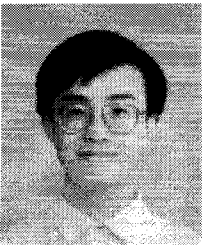
**Shiu-Sheng Yu** received the B.S. degree in Computer Science and Information Engineering from National Taiwan University in 1989 and M.S. degree in Computer Science from SUNY at Stony Brook in 1992. He is currently working towards the Ph.D.

degree in Computer Science and Information Engineering at National Taiwan University. His research interests include multimedia information systems and distributed systems.



**Jau-Hsiung Huang** received the B.S. degree in Electrical Engineering from National Taiwan University, Taiwan, R.O.C. in 1981 and the M.S. and Ph.D. degrees in Computer Science from the University of California, Los Angeles, CA, U.S.A., in 1985 and

1988, respectively. Since 1988, he has been a member of the faculty in the Department of Computer Science and Information Engineering, National Taiwan University, where he is currently a professor. He co-founded the Communications and Multimedia Laboratory in the department in 1990 to launch series of research in the areas of distributed multimedia systems and virtual reality. He has published over 40 technical papers in the areas of multimedia networking, high-speed networking, parallel and distributed systems and performance evaluation of computing systems.



**Ing-Chau Chang** received his B.S. degree in Department of Computer and Information Science from National Chiao-Tung University, Hsin-Chu, Taiwan, R.O.C., in 1990. and the M.S. and Ph.D. degrees in Institute of Computer Science and Information Engineering from

National Taiwan University, Taipei, Taiwan, R.O.C., in 1992 and 1997, respectively. His current research topics include computer-supported cooperative work, distance learning, multimedia network protocols, and multimedia systems.