1. Abstract

Learning is essential for everyone. In the past, we learned in school. With advances in Internet technology and multimedia technology, we can make cyber campus possible. Cyber campus is a virtual environment that simulates a real campus. In cyber campus, instructors and students can have their lesson as in the past, but now through Internet. It provides an alternative style of teaching and learning for them.

Our project, titled "Wireless Cyber Campus", is aimed at implementing such environment for instructors and students. In this report, we will mention the aim of our project, the functionality it provides and the implementation details of the system.

2. Introduction

2.1 Cyber Campus

Internet is an essential tool nowadays. We use Internet to retrieve information, communication by emails, newsgroup in last few years. As the advance in Internet technology and multimedia technology, we can make online-shopping, entertainment, video-on-demand and education possible through Internet now.

Students have great interests on this technology. They are keen in communicating through Internet by ICQ, in finding information through Internet. Today, we have applied Internet technology in education. Most University course has course homepage for delivering notes and relevant information through World-Wide-Web. Students can find the information they needed through Internet. However, is it enough for online-education? Can we further apply this Technology on the Internet?

In our project, we want to implement a Cyber Campus. I would like to introduce what is a cyber campus first. It is a virtual environment using Internet Technology to simulate a campus environment. Inside Cyber Campus, student and instructor can communicate with each other. They have their lesson as in the past, but now through Internet. This allows student can have lesson at anywhere. They no longer need to travel between schools and home. They can even take courses from overseas University.

Apart from these, students can have their own pace of learning. That means they no longer need to follow the pace of teaching of student. It is important that allowing students adjust their learning pace based on their ability, rather than controlling by the instructor. If they feel the pace of teaching of instructor so fast that they cannot follow, they can have the lesson later. Student can scheduled their own timetable, they can have lesson at anytime.

Comparing with the traditional campus, only 30-40 students crowded in a

classroom. Inside Cyber Campus, a lecture can accommodate over hundreds of students, as there is no room constrained anymore. Besides, teachers and students can enjoy closer interactions and better access to knowledge resources through Internet. Not only students will feel more open and comfortable when they ask questions with teachers, but also teachers can interact with the students more privately and directly, thus creating more personal relationship between teachers and students and improving the quality of education

In summary, Cyber Campus provides another way of learning for students and an inter-active way of communication for instructor and students. We hope that such an environment can help students to learn in an efficient and interesting way.

2.2 Project Objective

Our project aims to implement a Cyber Campus as mentioned above. In our project, we will implement a system, which includes the following functions:

- To make lecture through Internet possible, we would implement live-video streaming function, so that students can also attend lessons on-line.
- To enhance communication between instructor and students, we would implement video-conferencing, so that they can discuss more closely and privately.
- To make course materials to be delivered on-line, we would implement a file server for which students can search for relevant material to download.
- To allow personal scheduled timetable possible, we would implement video capturing and uploading, the captured lecture video would be uploaded to the video server so that students enjoy Video-On-Demand service to have their lesson.
- To enable students to open the course materials on the file-server through our system, we would embed an Acrobat Reader inside our system.

2.3 Project Overview

In our project, we will implement the whole system, system level and application level.

In system level, there are a server and libraries. The server works as a central unit of the whole system. It controls every message to pass through the system and controls the usage of resource inside the system. This is the most important part for the system to work properly.

The application level, programs are used by the instructor/students. They can invoke one or more activities (like message and video conference) in it. The client program will create an instance for each activity locally and every change will be distributed to the others.

The client program is used for received the response from the user and send it to the server. After server received their response, it will calculate the new state of the user and distribute it to all of the users within the system.

The libraries help us to develop a highly interactive network application. The libraries include Java SDK 1.2.2 and Java Media Framework (JMF) library. Based on them, we can design and build the system more efficient. Also, we have tried to develop the software into two approaches. One of the approaches is to have user-friendly interface. The second one is to have a robust system for client to use.

2.4 Programming Platform

All of the programs are written in Java. For the client/server program, it can run on Microsoft Windows and Unix platform with Java Media Framework installed.

Here is the Java architecture overview:

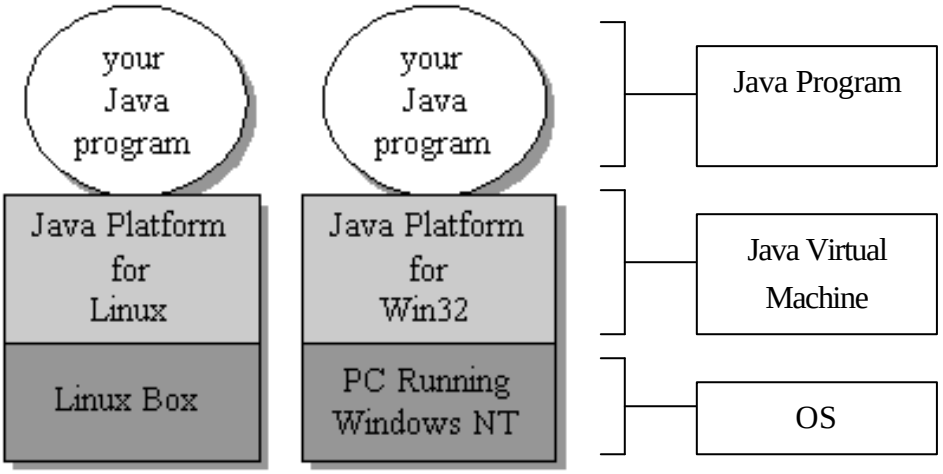


figure2.1 Java architecture Overview

2.5 Project Architecture

This project Cyber Campus is mainly divided into two parts.

The first part is the server, which mainly responsible for central control of the whole system. The second part is the client program, which is mainly responsible for display the learning material and interactive activities.

Basically, our work includes building the server and building the client program.

For the server we had built, they will be talked in details at Chapter 4 in this report.

For the client program we had built, detail will be talk in Chapter 5 in this report.

3. Background

3.1 Network

3.1.1 WinSock library

What is WinSock?

Windows Sockets (Winsock) specifies a programming interface based on the familiar "socket" interface from the University of California at Berkeley. It includes a set of extensions designed to take advantage of the message-driven nature of Microsoft Windows. Version 1.1 of the Windows Sockets specification was released in January 1993, and version 2.2.0 was published in May of 1996. Windows CE supports Winsock version 1.1.

Sockets are a general-purpose networking API. Winsock is designed to run efficiently on Windows operating systems while maintaining compatibility with the Berkeley Software Distribution (BSD) standard, known as Berkeley Sockets.

Windows Sockets applications can also be used in Windows CE, thus WinSock is suitable for design multi-platform software to provide generic network services.

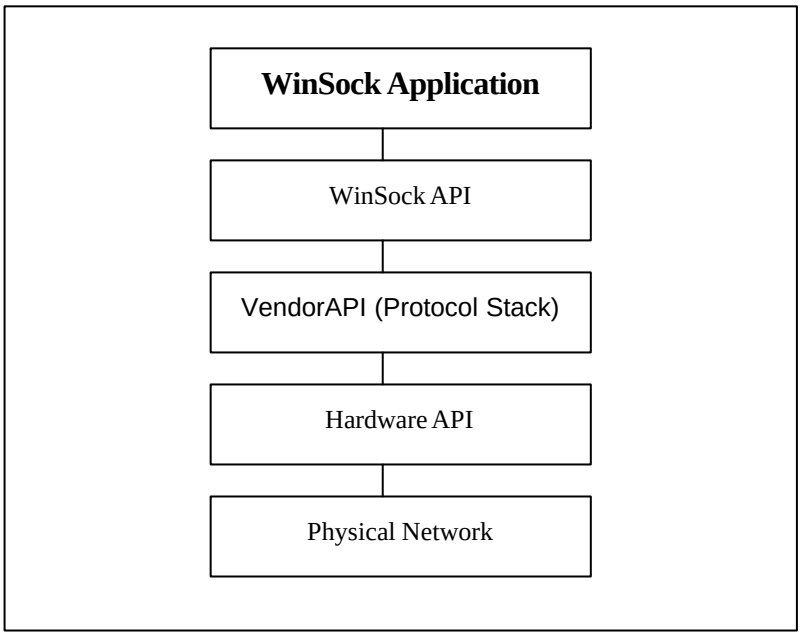


figure3.1 Overviews of how WinSock works

WinSock 2 Architecture

WinSock 2 has an all-new architecture that provides much more flexibility. The new WinSock 2 architecture allows for simultaneous support of multiple protocol stacks, interfaces, and service providers. There is still one DLL on top, but there is another layer below, and a standard service provider interface, both of which add flexibility.

WinSock 2 adopts the Windows Open Systems Architecture (WOSA) model, which separates the API from the protocol service provider. In this model the WinSock DLL provides the standard API, and each vendor installs its own service provider layer underneath. The API layer "talks" to a service provider via a standardized Service Provider Interface (SPI), and it is capable of multiplexing between multiple service providers simultaneously. The following sketch illustrates the WinSock 2 architecture.

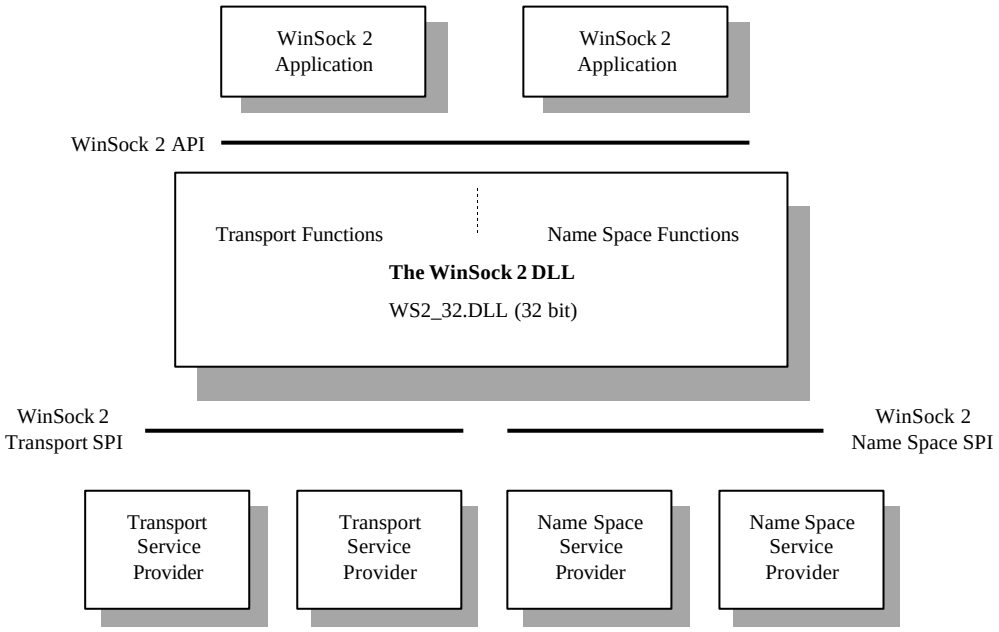


figure 3.2 WinSock 2 Architecture

Note that the WinSock 2 specification has two distinct parts: the API for application developers, and the SPI for protocol stack and namespace service providers. Notice also that the intermediate DLL layers are independent of both the application developers and service providers. These DLLs are provided and maintained by Microsoft and Intel. And lastly, notice that the *Layered Service Providers* would appear in this illustration one or more boxes on top of a transport service provider.

However, WinSock 2 is designed only for 32-bit Windows platforms; it cannot be run using Win32s on 16-bit Windows platforms. Nonetheless, as described earlier, (almost) all 16-bit WinSock 1.1 applications can be used. Windows NT version 4 has been called the "shell-update," since the most obvious change was the addition of the Windows 95 user interface. But there's a whole lot more that was changed within the NT4 kernel design. Significant modifications were done to make WinSock 2 the native network API for Windows NT4. As a result, *WinSock 2 will not be available for NT 3.51 or earlier*.*

Sockets Programming Paradigm under Windows

In WinSock 2, all connectionless protocols use SOCK_DGRAM sockets and all connection-oriented protocols use SOCK_STREAM sockets. Programmers should no longer rely on socket type to describe all of the essential attributes of a transport protocol.

Windows Sockets 2 takes the socket paradigm considerably beyond what it's original designers contemplated. As a consequence, a number of new functions have been added, all of which are assigned names that are prefixed with "WSA". In all but a few instances these new functions are expanded versions of an existing function from BSD sockets. The need to retain backwards compatibility mandates that we retain both the "just plain" BSD functions and the new "WSA" versions. The usage of functions for using a connection-oriented and connectionless application is shown in figure3.6 and figure3.7 respectively.

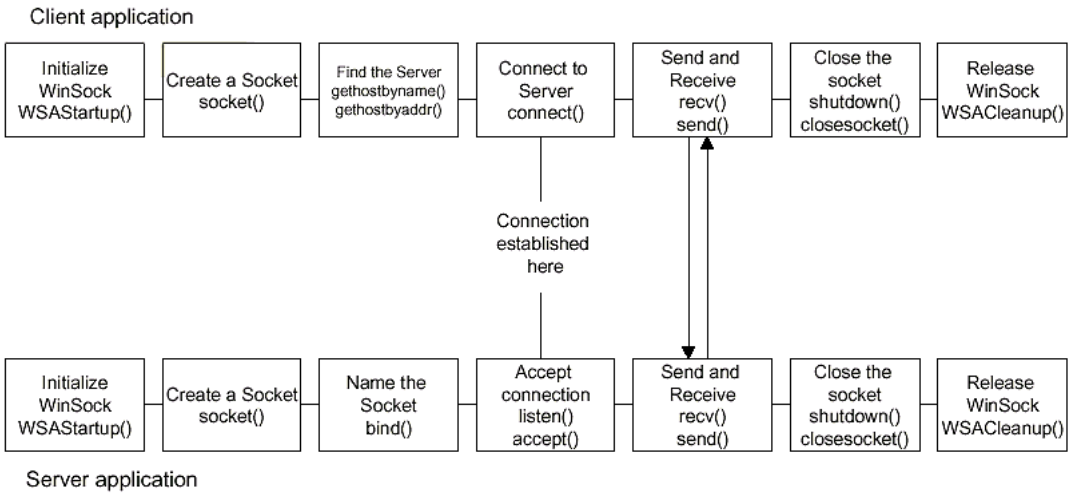


figure 3.3 Overview of Connection-Oriented Application

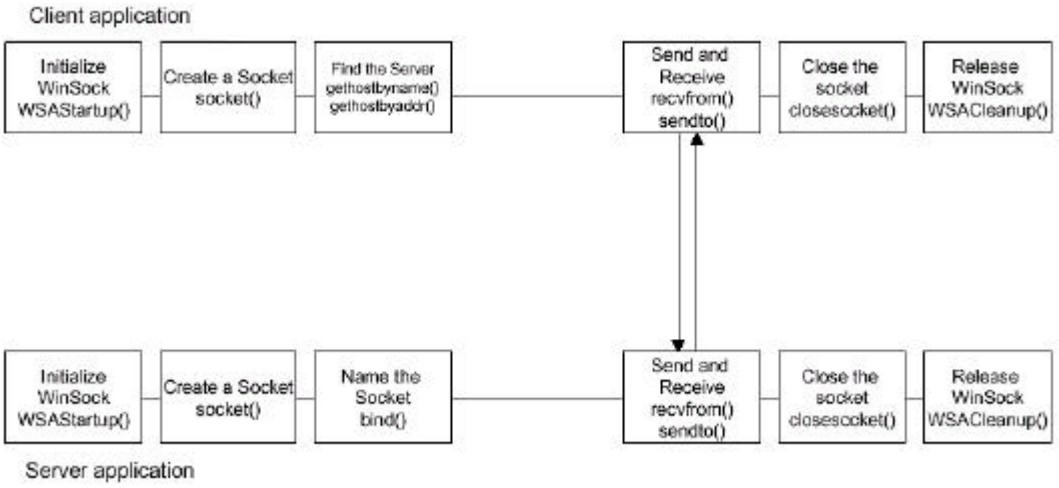


figure 3.4 Overview of Connectionless Application

Asynchronous Notification Using Event Objects

Introducing overlapped I/O requires a mechanism for applications to unambiguously associate send and receive requests with their subsequent completion indications. In WinSock 2 this may be accomplished via event objects that are modeled after Win32 events. WinSock event objects are fairly simple constructs, which can be created and closed, set and cleared, waited upon and polled. Their prime usefulness comes from the ability of an application to block and wait until one or more event objects become set.

Applications use WSACreateEvent() to obtain an event object handle which may then be supplied as a required parameter to the overlapped versions of send and receive calls (WSASend(), WSASendTo(), WSARecv(), WSARecvFrom()). The event object, which is cleared when first created, is set by the transport providers when the associated overlapped I/O operation has completed (either successfully or with errors). Each event object created by WSACreateEvent() should have a matching WSACloseEvent() to destroy it.

Event objects are also used in WSAEventSelect() to associate one or more FD_XXX network events with an event object.

The WSAEventSelect() and WSAEnumNetworkEvents() functions are provided. WSAEventSelect() behaves exactly like WSAAsyncSelect()

except that, rather than cause a Windows message to be sent on the occurrence of an FD_XXX network event (e.g.. FD_READ, FD_WRITE, etc.), an application-designated event object is set.

3.1.2 Java.net Package

What is Java.net Package?

Java.net package is used to provide the classes for implementing networking applications in Java. Using the socket classes, you can communicate with any server on the Internet or implement own Internet server. A number of classes are provided to make it convenient to use Universal Resource Locators (URLs) to retrieve data on the Internet. This package is developed since JDK1.0.

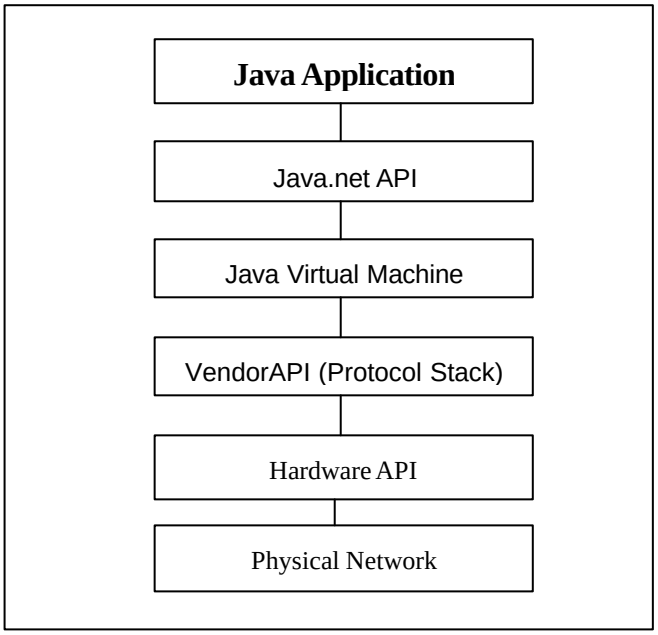


figure3.5 Overviews of how Java.net package works

Why Networked Java?

Java is the first programming language designed from the ground up with networking in mind. Java provides solutions to a number of problems – platform independence and security. It makes writing network programs easy. It's far easier to write network programs in Java than almost any other language. It's easy for Java application to send and receive data across the Internet.

Java Socket Class

A Socket is a connection between two hosts. It can perform seven basic operations:

- Connect to a remote machine
- Send data
- Receive data
- Close a connection
- Bind to a port
- Listen for incoming data
- Accept connections from remote machines on the bound port

Java's **Socket** class, which is used by both clients and servers, has methods that correspond to the first four of these operations. The least three operations are needed only by servers, which wait for clients to connect to them. They are implemented by the **ServerSocket** class. Java programs normally use client sockets in the following fashion:

1. The program creates a new **Socket()** constructor.
2. The socket attempts to connect to the remote host.
3. Once the connection is established, the local and remote hosts get input and output streams from the socket and use those streams to send data to each other. This connection is full-duplex; both hosts can send and receive data simultaneously.
4. When the transmission of data is complete, one or both sides close the connection.

3.2 Video Streaming and Capturing

3.2.1 Direct Show Library

What is DirectShow?

Microsoft® DirectShow® is an architecture for streaming media on the Microsoft® Windows® platform. It is based on the Component Object Model (COM). DirectShow provides for high-quality capture and playback of multimedia streams. It supports a wide variety of formats, including Advanced Streaming Format (ASF), Motion Picture Experts Group (MPEG), Audio-Video Interleaved (AVI), MPEG Audio Layer-3 (MP3), and WAV files.

It supports capture using Windows Driver Model (WDM) devices or older Video for Windows devices..

DirectShow simplifies media playback, format conversion, and capture tasks. At the same time, it provides access to the underlying stream control architecture for applications that require custom solutions.

Examples of the types of applications

- DVD players
- Video editing applications
- AVI to ASF converters
- MP3 players
- Digital video captures applications.

DirectShow System Overview

The following diagram shows the relationship between an application, the DirectShow components, and some of the hardware and software components that DirectShow supports.

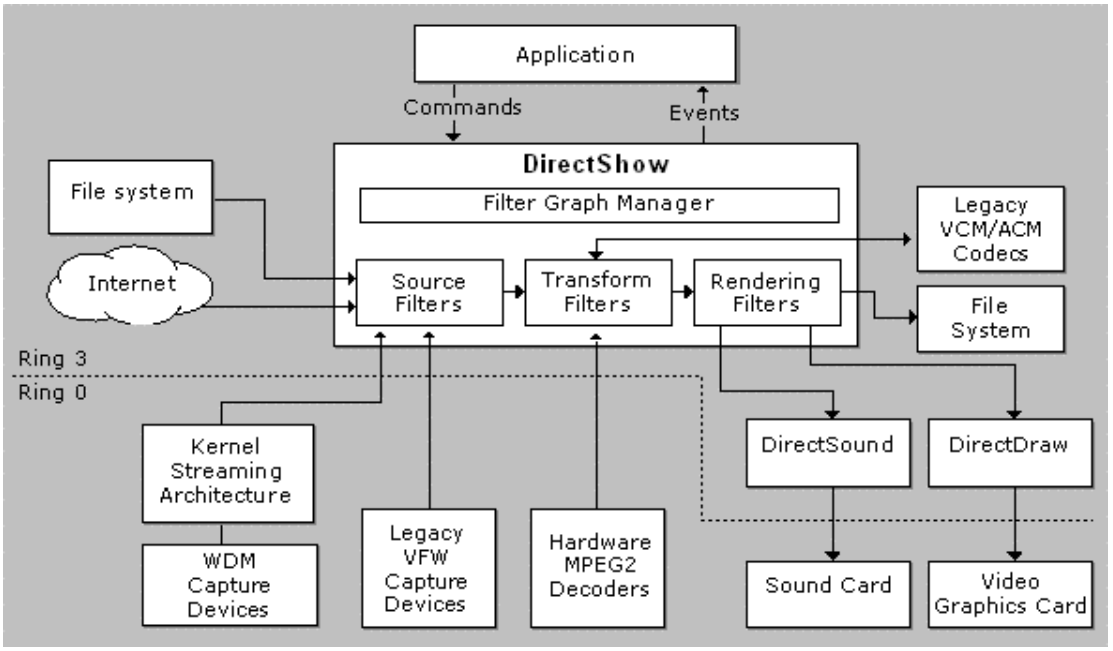


figure 3.6 DirectShow System Overview

As illustrated here, DirectShow enables applications to play files and streams from various sources, including local files, local CD and DVD drives, remote files on a network, newer TV-tuner and video capture cards based on the Windows Driver Model (WDM), and legacy Video For

Windows® video capture cards. DirectShow has native compressors and decompressors for some file formats, and many third-party hardware and software decoders are compatible with DirectShow. In addition, DirectShow supports legacy VfW codecs based on the Video Compression Manager (VCM) and Audio Compression Manager (ACM) interfaces. Playback makes full use of DirectDraw hardware acceleration and DirectSound capabilities when the hardware supports it.

DirectShow Application Programming

At the heart of the DirectShow services is a modular system of pluggable components called filters, arranged in a configuration called a filter graph. A component called the filter graph manager oversees the connection of these filters and controls the stream's data flow.

✍✍ **Filter**

The basic building block of DirectShow is a software component called a filter. A filter generally performs a single operation on a multimedia stream. For example, there are DirectShow filters that

- Read files.
- Get video from a video capture device.
- Decode a particular stream format, such as MPEG-1 video.
- Pass data to the graphics or sound card.

✍✍ **Filter Graph**

A filter graph is composed of a collection of filters of different types. Most filters can be categorized into one of the following three types.

- A *source filter*, which takes the data from some source, such as a file on disk, a satellite feed, an Internet server, or a VCR, and introduces it into the filter graph.
- A *transform filter*, which takes the data, processes it, and then passes it along.
- A *rendering filter*, which renders the data; typically this is rendered to a hardware device, but could be rendered to any location that accepts media input (such as memory or a disk file).

Filters receive input and produce output. For example, if a filter decodes MPEG-1 video, the input is the MPEG-encoded stream and

the output is an uncompressed RGB video stream.

To perform a given task, an application connects several filters so that the output from one filter becomes the input for another. A set of connected filters is called a **filter graph**. As an illustration of this concept, the following diagram shows a filter graph for playing an AVI file.

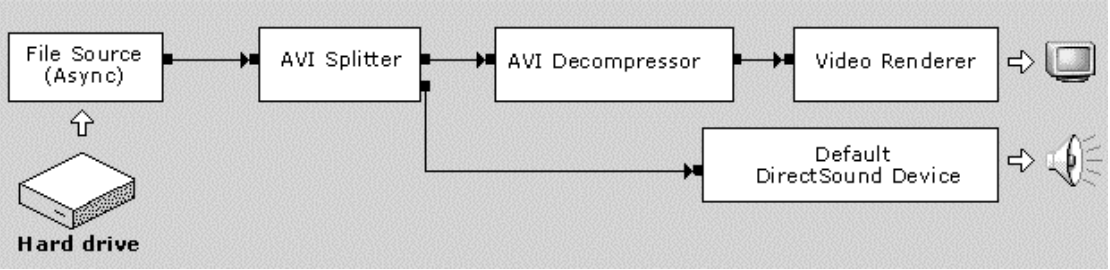


figure 3.7 DirectShow filter graph

✍✍ **Filter Graph Manager**

DirectShow provides a high-level component called the Filter Graph Manager. The Filter Graph Manager controls the flow of data through the graph. We can makes high-level API calls such as "Run" (to move data through the graph) or "Stop" (to stop the flow of data). We can also access the filters directly through COM interfaces. The Filter Graph Manager also passes event notifications to the application, so that our application can respond to events, such as the end of a stream.

In addition, the Filter Graph Manager simplifies the process of building a filter graph. For example, you can specify a file name, and the Filter Graph Manager will build a graph to play that file.

✍✍ **Writing a DirectShow Application**

A typical DirectShow application performs three basic steps, as illustrated in the following diagram.

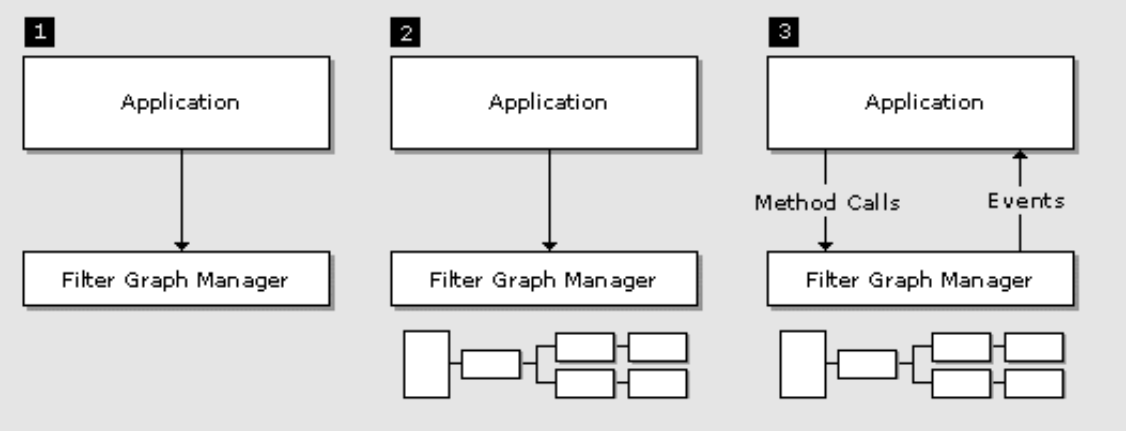


figure 3.8 Process of writing a DirectShow application

1. Creates an instance of the Filter Graph Manager, using the CoCreateInstance function.
2. Uses the Filter Graph Manager to build a filter graph.
3. Controls the filter graph and responds to events.

3.2.2 Java Media Framework (JAVA JMF)

The Java Media Framework (JMF) is an application-programming interface (API) for incorporating time-based media into Java applications and applets. It provides supports for

- Capturing and storing media data
- Controlling the type of processing that is performed during playback
- Performing custom processing on media data streams.
- Streaming and conferencing applications
- Providing access to raw media data
- Enable the development of custom, downloadable demultiplexers, codecs, effects processors, multiplexers, and renderers (JMF *plug-in*s)

Why using JMF?

JMF support **Real-Time Transport Protocol** (RTP) as a solution to send or receive a live media broadcast or construct a videoconference over the Internet or intranet.

Streaming Media

When media content is streamed to a client in real-time, the client can begin to play the stream without having to wait for the complete stream to download. In fact, the stream might not even have a predefined duration downloading the entire stream before playing it would be impossible. The term streaming media is often used to refer to both this technique of delivering content over the network in real-time and the real-time media content that's delivered.

Streaming media is everywhere you look on the web-live radio and television broadcasts and webcast concerts and events are being offered by a rapidly growing number of web portals, and it's now possible to conduct audio and video conferences over the Internet. By enabling the delivery of dynamic, interactive media content across the network, streaming media is changing the way people communicate and access information.

Real-Time Transport Protocol

RTP provides end-to-end network delivery services for the transmission of real-time data. RTP is network and transport-protocol independent, though it is often used over UDP.

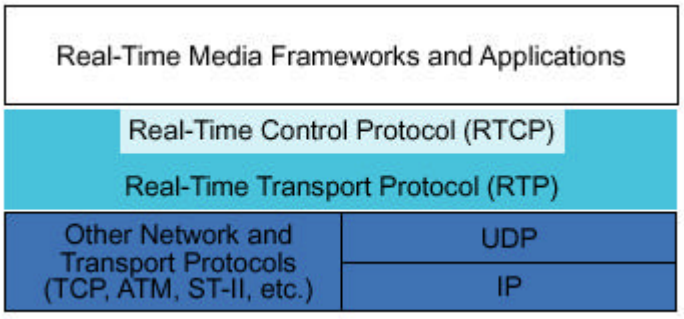


figure3.9 RTP architecture

RTP can be used over both unicast and multicast network services. Over a *unicast* network service, separate copies of the data are sent from the source to each destination. Over a *multicast* network service, the data is sent from the source only once and the network is responsible for transmitting the data to multiple locations. Multicasting is more efficient for many multimedia applications, such as videoconferences. The standard Internet Protocol (IP) supports multicasting.

RTP Applications

RTP applications are often divided into those that need to be able to receive data from the network (RTP Clients) and those that need to be able to transmit data across the network (RTP Servers). Some applications do both -- for example, conferencing applications capture and transmit data at the same time that they' re receiving data from the network.

Receiving Media Streams from network

Being able to receive RTP streams is necessary for several types of applications. For example:

- Conferencing applications need to be able to receive a media stream from an RTP session and render it on the console.
- A telephone answering machine application needs to be able to receive a media stream from an RTP session and store it in a file.
- An application that records a conversation or conference must be able to receive a media stream from an RTP session and both render it on the console and store it in a file.

Transmitting Media Streams across network

RTP server applications transmit captured or stored media streams across the network.

For example, in a conferencing application, a media stream might be captured from a video camera and sent out on one or more RTP sessions. The media streams might be encoded in multiple media formats and sent out on several RTP sessions for conferencing with heterogeneous receivers. Multiparty conferencing could be implemented without IP multicast by using multiple unicast RTP sessions.

3.2.3 Comparison

Both DirectShow and JMF can perform video playback, capturing and streaming. The underlying architectures for both API are quite similar. DirectShow uses Filter Graph and JMF uses JMF player/process model. The architecture connects different components to perform certain task.

For instance, in writing a simple MPEG playback application. DirectShow connects the source filter, which get the data from a MPEG file, to the splitter filter and decompression filter. Finally a renderer is used to display the video content. JMF use a datasource to read the video data and a player performs split, decompress and rendering of the video. Comparing to DirectShow architecture, JMF is much simpler as there are only two main components there. But the underlying principle is the same for both.

For video playback, DirectShow supports more video formats than the JMF does. Program written in DirectShow supports formats such as MPEG-2 which JMF does not support. By installing the suitable codec in the Windows platform, DirectShow can playback the format. JMF doesn't support much video formats currently, it only supports MPEG1 and AVI format.

In terms of performance, we experience that DirectShow has better performance than JMF. The overhead to start the system for DirectShow is much smaller than that of JMF. The delay is particularly significant for video capturing. Besides, as Java is work under the Java Virtual Machine, the overall performance is also poorer than DirectShow.

To implement video streaming, DirectShow provides a mechanism for programmers to implement a source filter to stream and receive video. Programmers are required to connect the existing filter to the source filter. JMF provides two methods to perform video streaming. One is using RTP as we have mentioned before and other one is implementing a datasource. Implementing a datasource is quite similar as the filter does. Programmer can implement a datasource that can be connected with the processor to perform video streaming.

In our system, we choose to use RTP instead of writing a datasource. The reason is we want to try using a different protocol that we haven't use before. We hope we can learn more about this new protocol during our project.

After using these two API, we found that using JMF is much easier than DirectShow. The reason is that JMF has clearer documentation and sample programs which demonstrate how to use the API, so that we can

implement the video streaming much easier. Besides, the Component Object Modal (COM) architecture used by DirectShow is more difficult to understand when comparing with JMF player/processor modal. COM requires a thorough understanding of the interface of the COM object and COM class. In JMF, it uses the Object-Oriented Principle which is easier to understand.

In conclusion, DirectShow has better performance and has comprehensive support of different video formats. However, it is more difficult to use and understand than JMF does. JMF provides two kinds of method for us to implement video streaming. We found that using RTP to implement is an easier method. So we recommend using Java JMF for implementation of video streaming.

3.3 User Interface

3.3.1 Microsoft Foundation Class (MFC)

MFC overview

The Microsoft Foundation Class Library (MFC) is an "application framework" for programming in Microsoft Windows. Written in C++, MFC provides much of the code necessary for managing windows, menus, and dialog boxes; performing basic input/output; storing collections of data objects; and so on. All you need to do is add your application-specific code into this framework. And, given the nature of C++ class programming, it's easy to extend or override the basic functionality the MFC framework supplies.

The MFC framework is a powerful approach that lets you build upon the work of expert programmers for Windows. MFC shortens development time; makes code more portable; provides tremendous support without reducing programming freedom and flexibility; and gives easy access to "hard to program" user-interface elements and technologies, like Active technology, OLE, and Internet programming. Furthermore, MFC simplifies database programming through Data Access Objects (DAO) and Open Database Connectivity (ODBC), and network programming through Windows Sockets. MFC makes it easy to program features like property sheets ("tab dialogs"), print preview, and floating, customizable toolbars.

Using the Classes to Writer User Interface

The classes in the Microsoft Foundation Class Library (MFC) make up an "application framework" — the framework on which we build an application for Windows. At a very general level, the framework defines the skeleton of an application and supplies standard user-interface implementations that can be placed onto the skeleton. We can just fill the rest of skeleton. We can use the Microsoft Visual C++™ resource editors to design your user-interface elements visually, ClassWizard to connect those elements to code, and the class library to implement your application-specific logic.

3.3.2 Java Foundation Class (JFC)

What Is the JFC?

The Java Foundation Classes software extends the original Abstract Window Toolkit (AWT) by adding a comprehensive set of graphical user interface class libraries. The rest of this page describes each new JFC release feature and tells you where to find more information about it.

JFC/Swing GUI Components

These components are written in the Java programming language, without window-system-specific code. This facilitates a customizable look and feel without relying on the native windowing system, and simplifies the deployment of applications.

Pluggable Look & Feel

This feature gives users the ability to switch the look and feel of an application without restarting it and without the developer having to subclass the entire component set.

3.4 Active X

Software Components are a natural evolution of Object Oriented software development, enabling the isolation of parts of an application into separate components. Such components can be shared between applications, and since components are only accessed through a rigidly defined interface, their implementation can be changed without impacting applications, which

use them.

Microsoft's widely used Component Object Model (COM) defines a binary standard for component integration, allowing COM components created using Visual BASIC (for example), to be accessed from an application created using Visual C++.

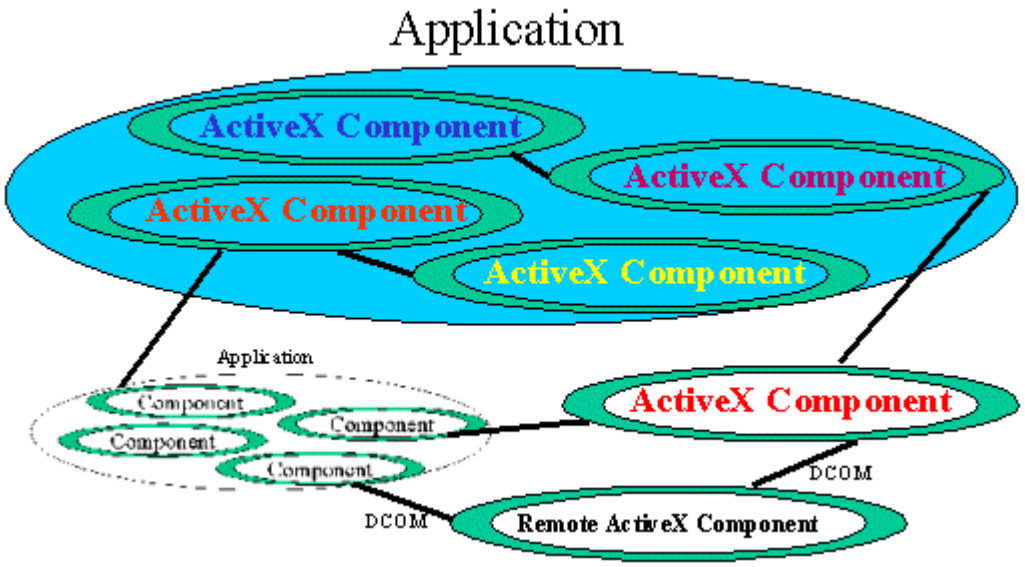


figure 3.10 Relationship between ActiveX component and application

Modern software design under Microsoft Windows practically mandates that an application be designed using an ActiveX Component based approach. One of the benefits of doing so is that parts of an application's functionality can be made accessible to other applications, not only the same host, but also remotely -- using Distributed COM (DCOM).

In order to invoke ActiveX object in Java, we have found a tool, J-Integra, to achieve this goal. J-Integra is a COM-Java bridging tool. Using J-Integra we can access ActiveX Components as though they were Java Objects, and we can access pure Java objects as though they were ActiveX Components. In the system, we use ActiveX object of Acrobat Reader and a Calendar. We add the reader so that we can open a PDF file in our system.

J-Integra works with any Java Virtual Machine, on any platform, and requires no native code (no DLLs). Below is architecture of how J-Integra communicates with JVM.

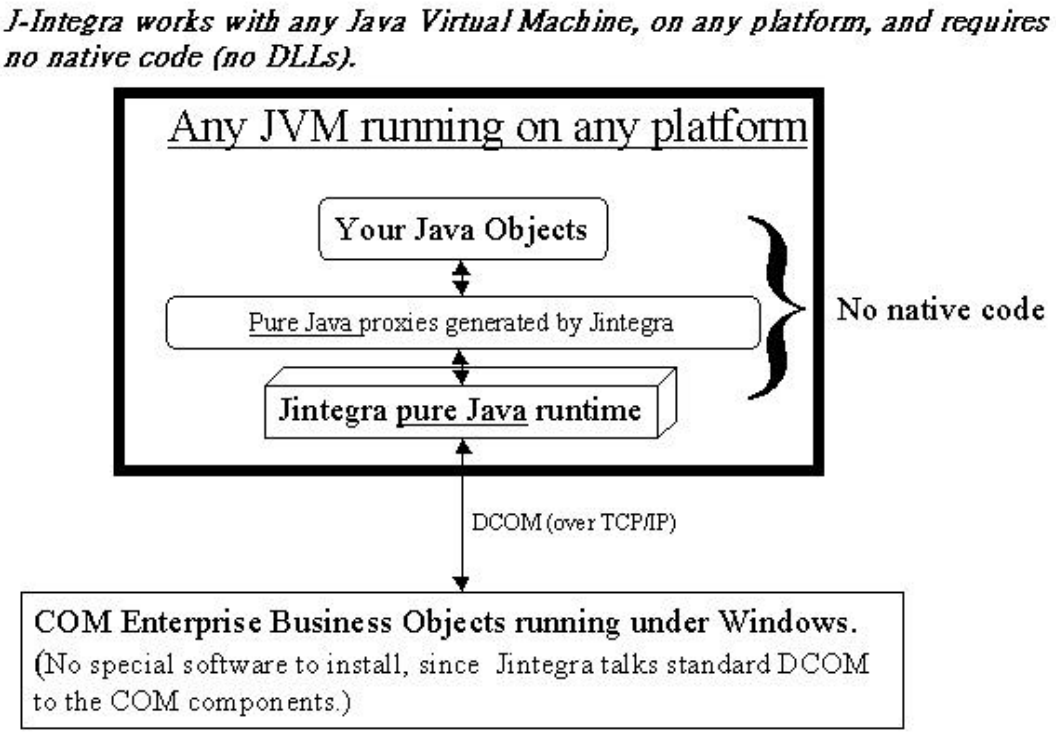


figure 3.11 Architecture J-Integra

J-Integra's pure Java runtime talks to COM components using Distributed COM (DCOM) layered over Remote Procedure Calls (RPC), which are layered on TCP/IP. So at the lowest level J-Integra uses the totally standard Java networking classes.

To the Java programmer, J-Integra makes COM components look just like Java objects, presenting COM properties, methods and events as Java properties, methods and events.

Here are basic steps to use an ActiveX object in Java

1. Generate the proxies used to access the ActiveX object using J-Integra.
2. Create and run the Java Application.

Through this tool, we can successfully invoke an ActiveX object in our system to enhance our functionality

4. Cyber Campus – Server

4.1 Introduction

Server is a centralized control of the whole system. It supports authentication of users. It provides three main kinds of services for clients, message passing server, file sharing server and video streaming server.

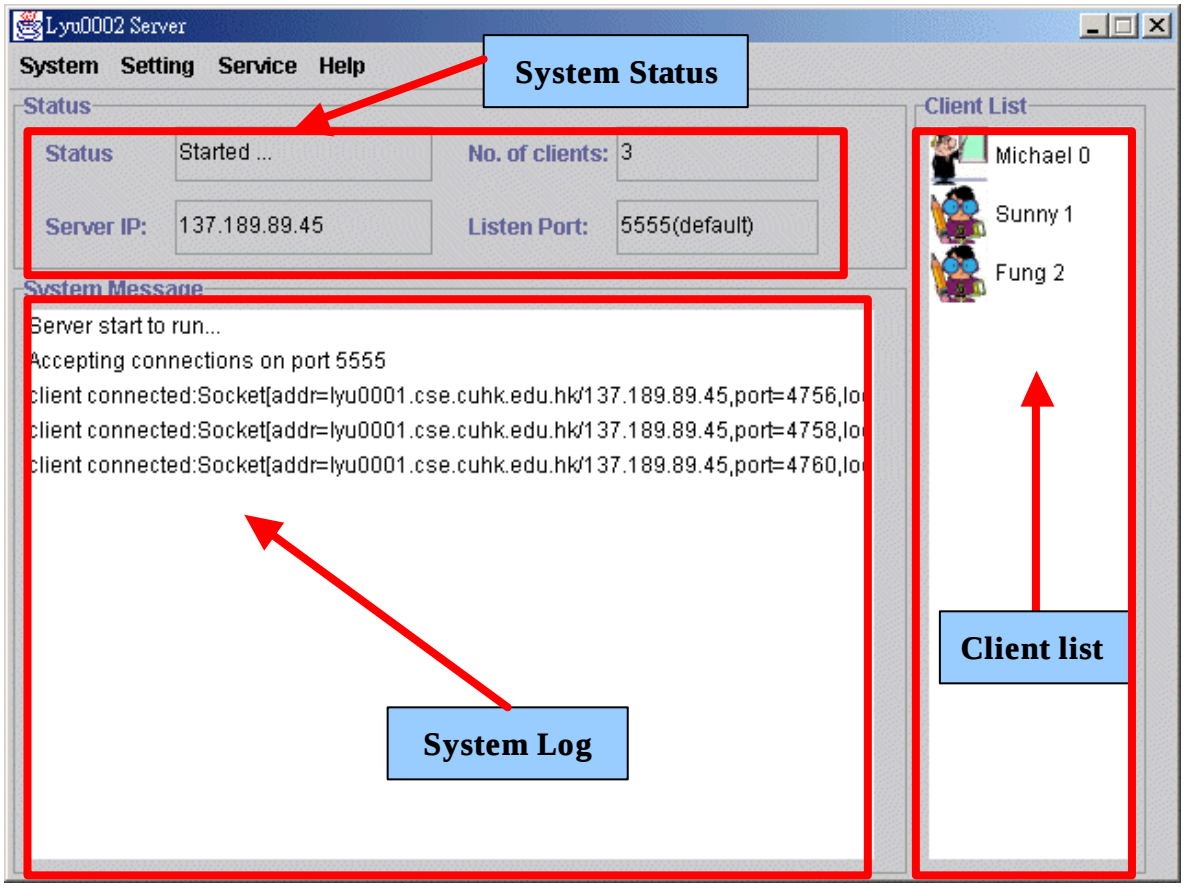


figure4.1 User interface of the server program

The features are system status, system log and client list.

System status

Server IP:	Show the IP address of the server.
Status:	Show the server' s status is running, stop or error.
No. of clients:	Show the total number of clients which had logged on.
Listen Port:	Show the port number that the server is listening at. By default the port number is 5555.

Login Client List

List that show the entire client name and its client id that are logged in the server. There are mainly two types of users in the system, instructor and student class. Instructor class users have more rights than student class users. Figure4.1 shows the icons for different kinds of users in our system.

System Message

This box is used to show all system log message, for example, client login/out, the type of error message of the server.

4.2 Menu

4.2.1 System Menu

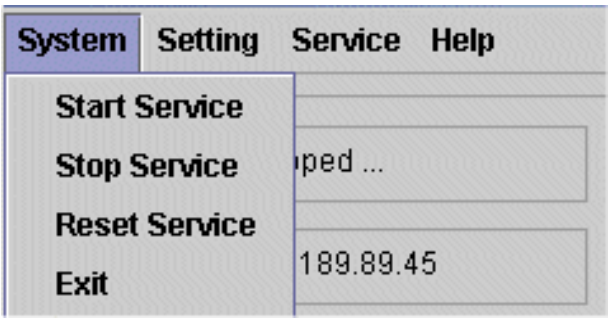


figure4.2 System menu of the server

Start Service

The server will start to run service. Client can be login the system for service.

Stop Service

The server will stop service. All the client will be forced to logout and the server close the socket for listening.

Reset Service

The server will stop service and start the service immediately. Because the server had down for a short time, all the connection between client and server will be lost.

Exit

Stop the service if the server is running. After close all the connection, the server program will exit.

4.2.2 Setting Menu

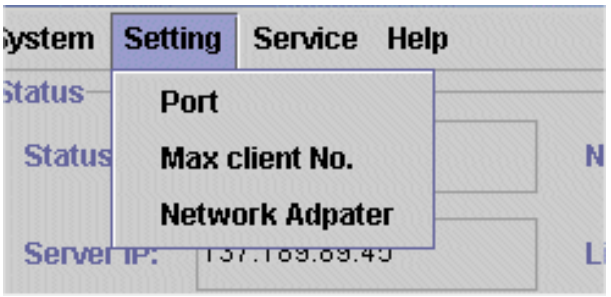


figure4.3 Setting menu of the server

Port

It used to change the port number for the server to listen at. When the service is started, the port number can' t be changed.

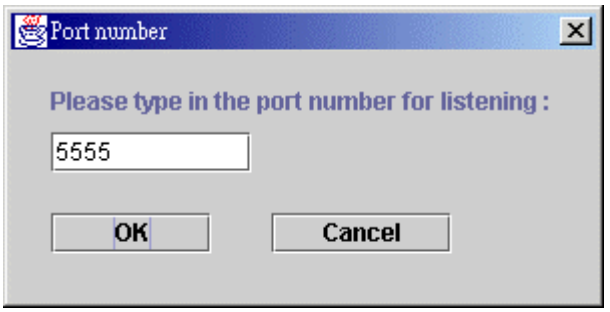


figure4.4 Port number

Max. client No.

It used to set the maximum number of client that can be login the system.

Network Adapter

This function is used for a server, which is multi-homed machine.

It can choose the network adapter for bind the listening socket.

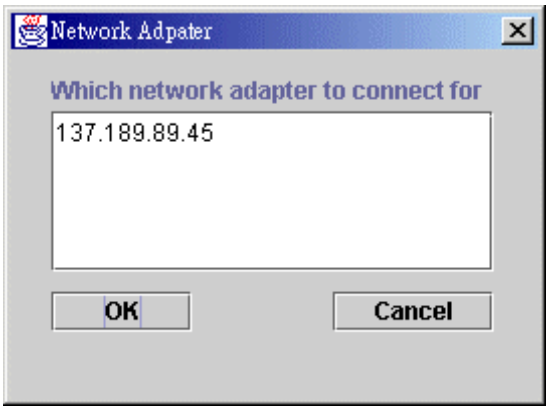
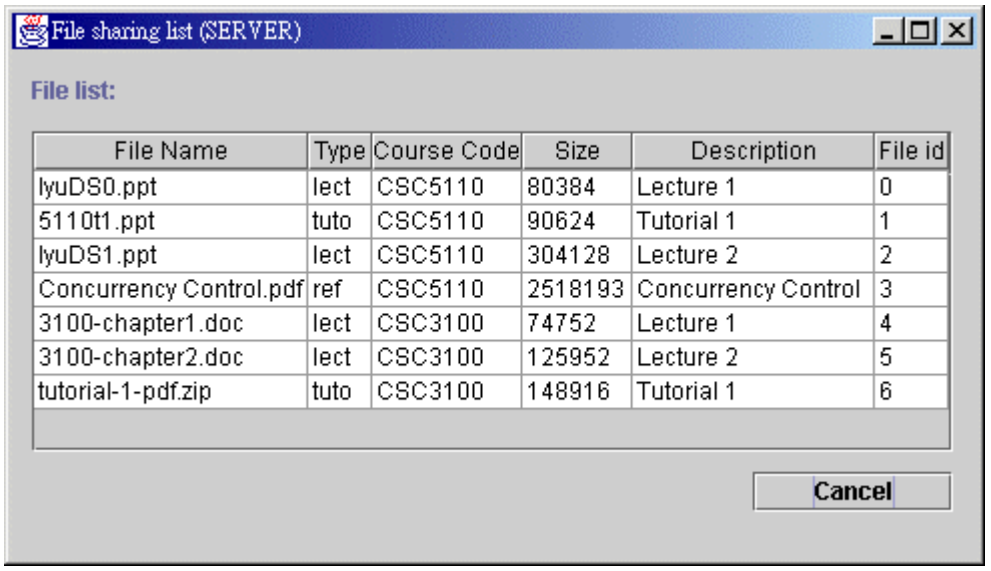


figure4.5 Network adapter

4.2.3 Service Menu

figure4.6 Service menu of the server

File Sharing Service



File Name	Type	Course Code	Size	Description	File id
lyuDS0.ppt	lect	CSC5110	80384	Lecture 1	0
5110t1.ppt	tuto	CSC5110	90624	Tutorial 1	1
lyuDS1.ppt	lect	CSC5110	304128	Lecture 2	2
Concurrency Control.pdf	ref	CSC5110	2518193	Concurrency Control	3
3100-chapter1.doc	lect	CSC3100	74752	Lecture 1	4
3100-chapter2.doc	lect	CSC3100	125952	Lecture 2	5
tutorial-1-pdf.zip	tuto	CSC3100	148916	Tutorial 1	6

figure4.7 The list of file sharing

It used to show the files and their details, which the server is sharing. The server acts as a file server to store all the course materials for students to download from it. Only instructor has the right to upload the material to the server. As shown above, there are mainly three main kinds of course materials, namely lect (Lecture), tuto (Tutorial) and ref (Reference).

Video-on-demand Service

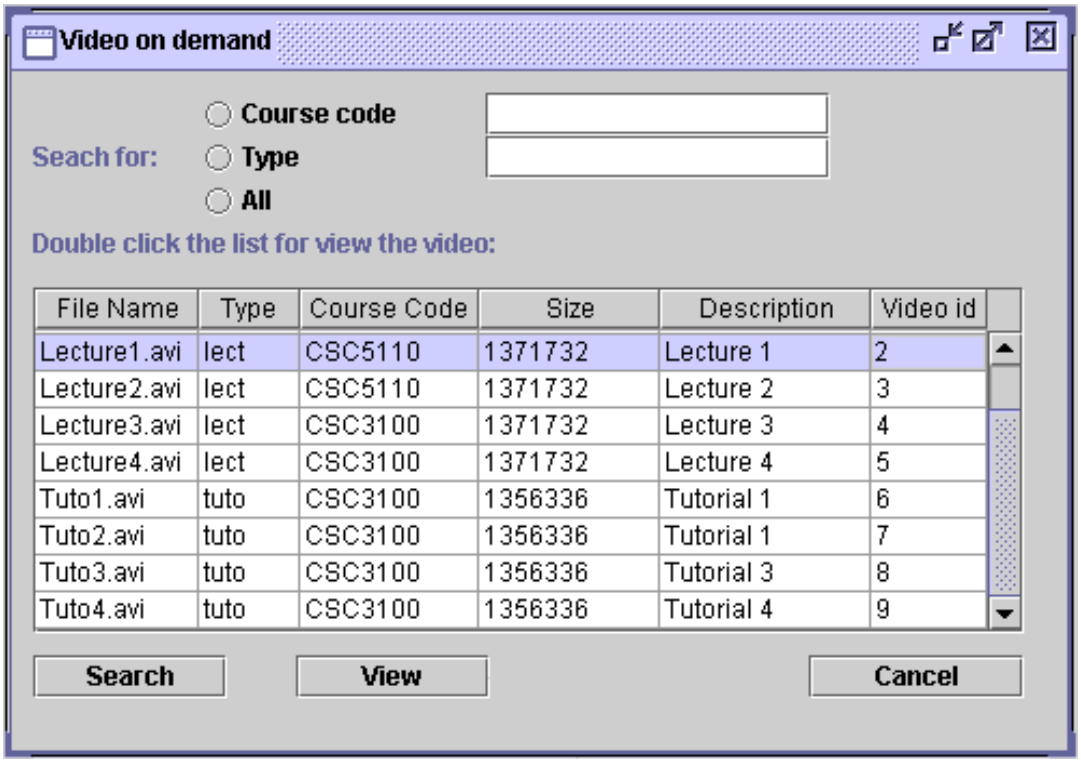


figure4.8 The list of video-on-demand

It used to show the video and their details, which server can give client to see the video when they want.

5. Cyber Campus - Client

5.1 Introduction

Client is the program for instructor and students. It supports functions of File Sharing, Live lecture, Video Capturing, Video Playback, Acrobat Reader and Video-On-Demand service for lecture purpose. Besides, client supports video-conferencing and chatting to provide an inter-active communication for instructor and students. With the client program, instructor and student can enjoy the benefits of the Cyber Campus.

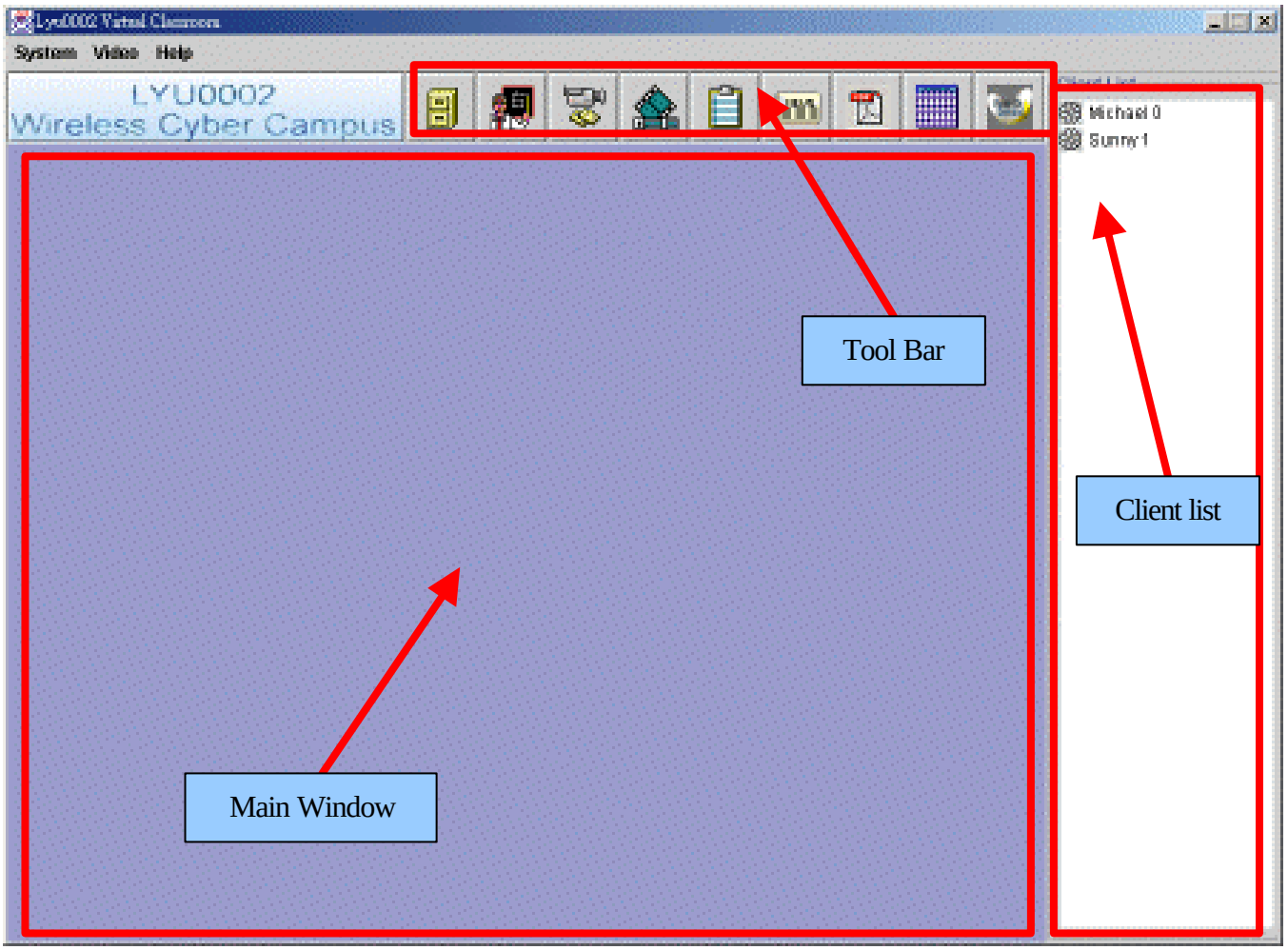


figure5.1 User interface of the client program

The user interface is divided into four components, which include tool bar, client list and main window of your system.

5.1.1 Login Client List

A list that show the entire client name and its client id that are logged in the server.

5.1.2 Main Window

This window is the most important for our system to show all the functionality.

Tool Bar

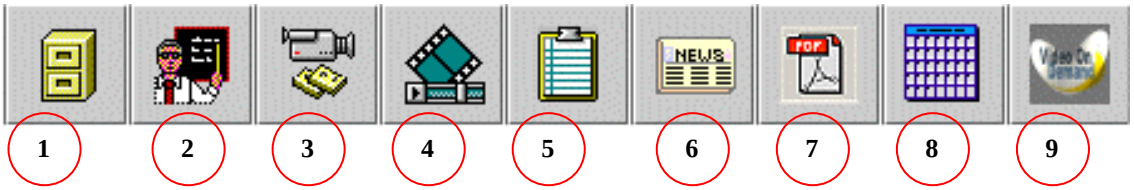


figure 5.2 Tool Bar

There are main nine functions for the client to choose in the tool bar:

1. **File sharing**
   - ✍✍ A file server to share files on the system
2. **Live lecture**
   - ✍✍ Attend the live lecture on the internet
3. **Preview/Capture web camera**
   - ✍✍ Preview the video or save the live video into the local hard disk
4. **Video playback**
   - ✍✍ Playback of local video file in the system
5. **Notice board**
   - ✍✍ See any announcement from the instructor
6. **Newsgroup**
   - ✍✍ Place for the student to discuss different topic
7. **PDF file reader**
   - ✍✍ A reader to open PDF document
8. **Calendar**
   - ✍✍ Calendar
9. **Video-on-demand**
   - ✍✍ Student can watch the video shared in the system whenever they want

5.1.3 Menu Bar

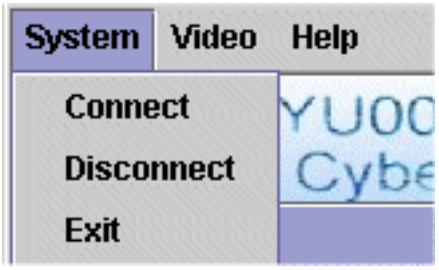


figure 5.3 Menu Bar

Connect

Make a connection to the server and login the system.

Disconnect

Logout the system and disconnect from the server.

Exit

Logout the system and disconnect from the server, then exit the program.

5.1.4 Client list event

figure 5.4 Event list for client

Send Message

A chatting function for student to communicate

Invite video conference

Invite other login student to have a video conference

5.2 Feature

5.2.1 Send Message

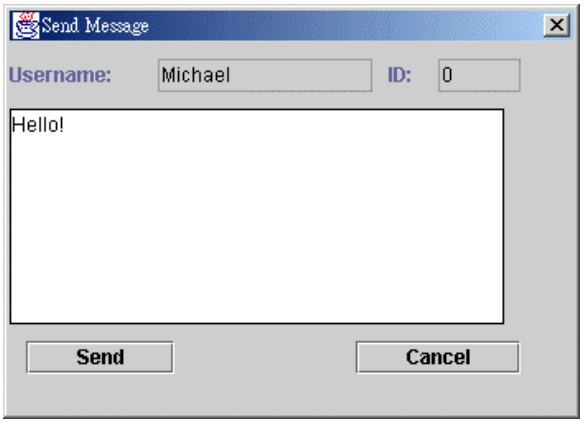


figure 5.5 Send message dialog

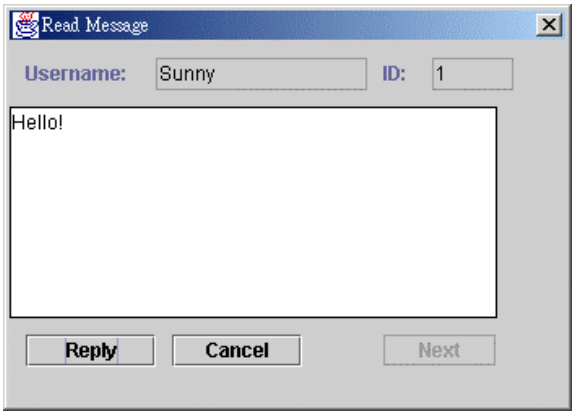


figure 5.6 Read message dialog

Introduction

Students like to chat with each other. This function provides a means for students to chat just like ICQ. As shown in the figure below, the user interface is quite similar to ICQ. This makes students feel more familiar with the system.

Using the Utilities

✍✍ **Send message**

There are two methods to invoke the send message event.

1. Double click the mouse button on the client' s name
2. Right mouse click the client's name and then click the send message item

After that, the send message dialog box (fig 5.5) will pop up for user to send message to that client. This can make the interface user-friendlier.

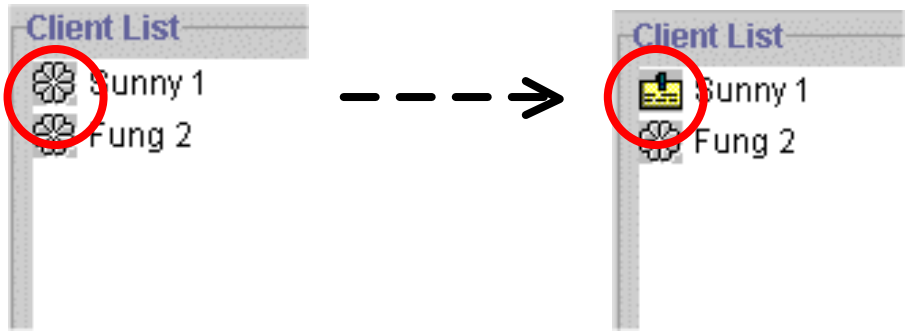


figure5.7 Snapshot of the receive message event

✍✍ **Receive message**

Receive message dialog box (fig 5.6) will not pop up immediately when message is incoming. The status, which is in front of the client' s name, will change. The snapshot is shown in figure 5.7. User can see the received message by double click the mouse button on the client' s name when the user wants to see the message.

5.2.2 File sharing 

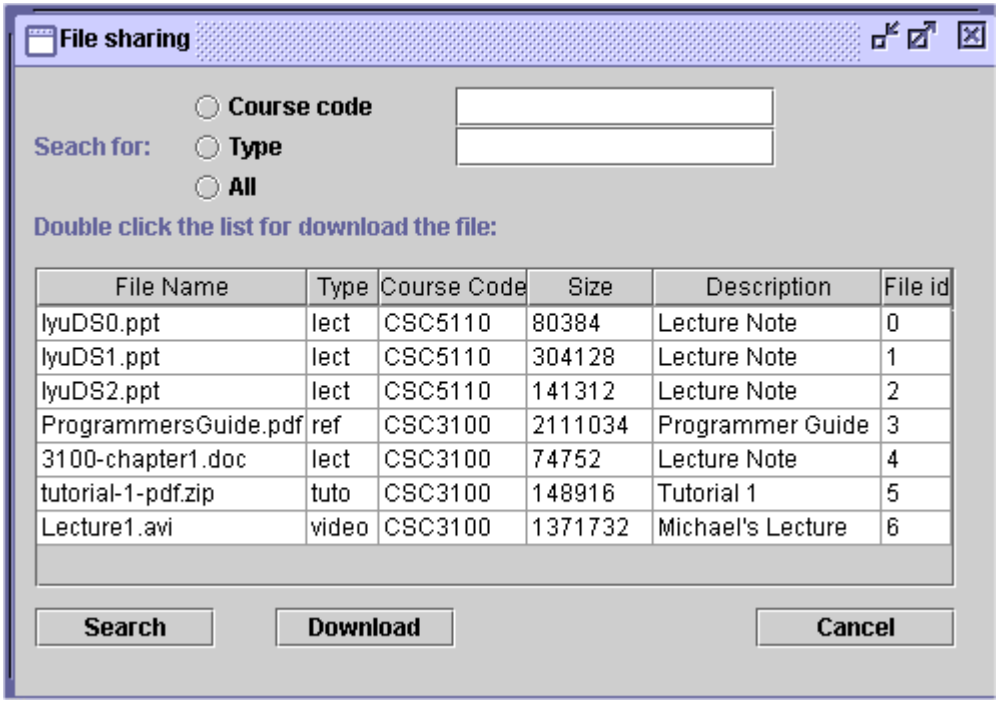


figure5.8 Snapshot of the file sharing

Introduction

In Cyber Campus, instructor can upload the course materials into the file server so that students can download the required material through the file sharing service.

Using the Utilities

Instructor and student will have different permission to use this function.

If the user is a instructor, (s)he can used both upload file to and download file from the server. If the user is a student only, (s)he can only download the file that shared in the system.

figure 5.9 file sharing menu

Upload file

Instructor can upload lecture notes, tutorial notes and any reference materials. They can write some description for each notes or materials so that student can choose what they want to use. All the files are stored in the file server.

Download file

After choosing the "Download file" menu or by clicking the file sharing icon for the instructor, the file sharing dialog (fig. 5.8) will be shown in the main window.

In the file sharing dialog, there are some features to help user to find the document which is useful and related to their study.

✍✍ Sorting

User can sort the files in ascending order by clicking the header of that column. After sorting, user can select the file for download easily.

✍✍ Filtering by course code

By using the filtering feature, user can use the filtering feature to filter the specified course code; figure 5.10 shows the results after filter by the course code.

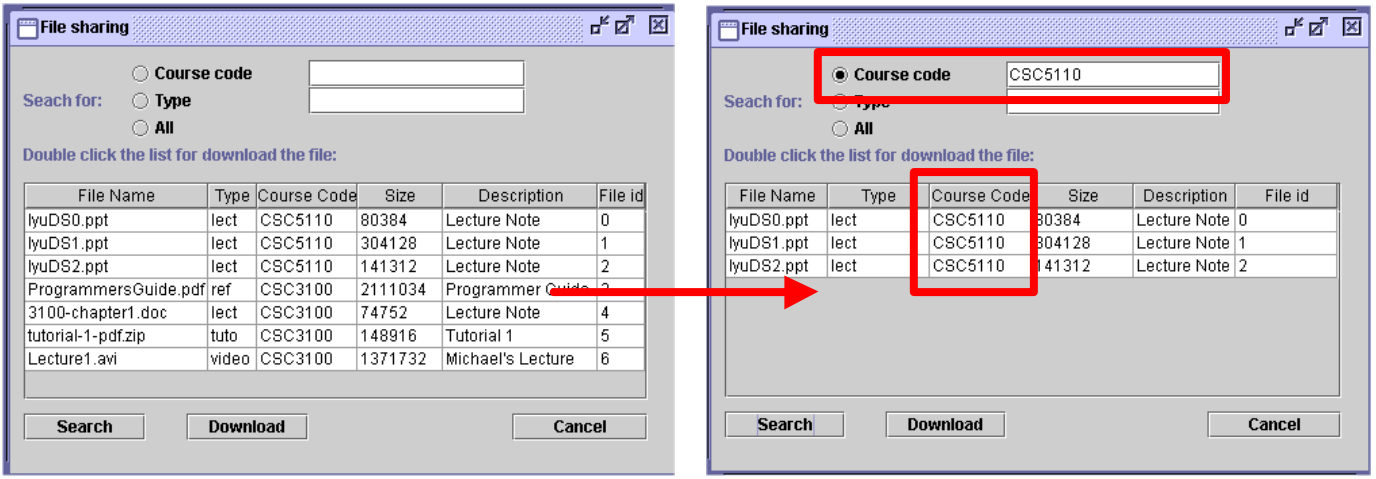


figure 5.10 Result of the file after filtering by the course code

✍✍ Filtering by file type

By using another filtering feature, user can filter the file by specified file type; figure 5.11 shows the results after filter by the file type.

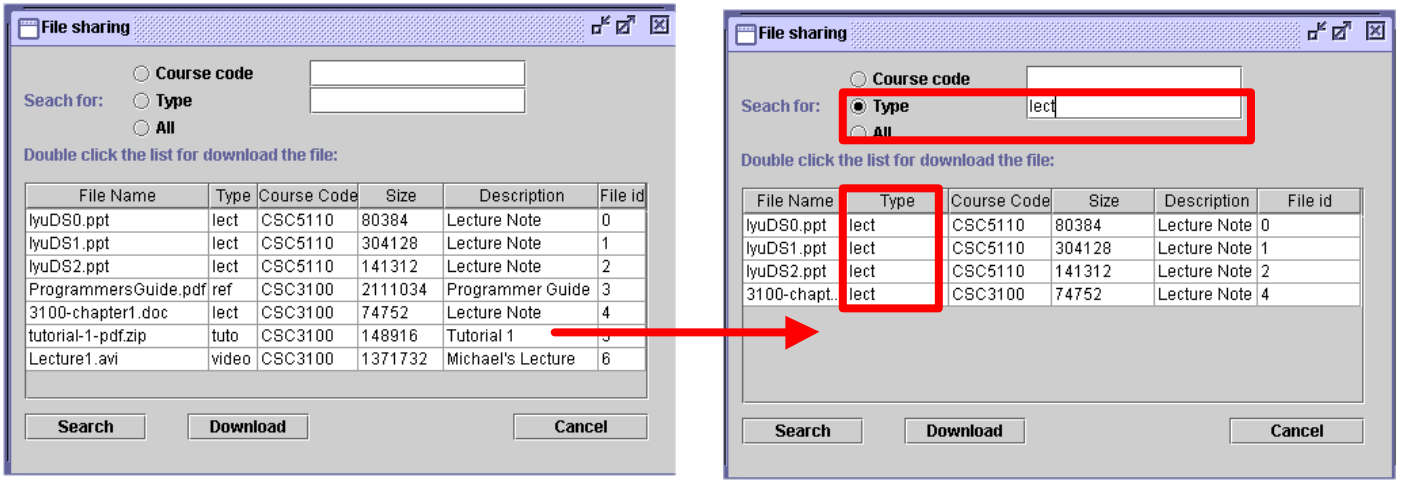


figure 5.11 Result of the file after filtering by file type

User can double click the file to download it or select the file first and then the download button in the file-sharing dialog.

5.2.3 PDF file reader 

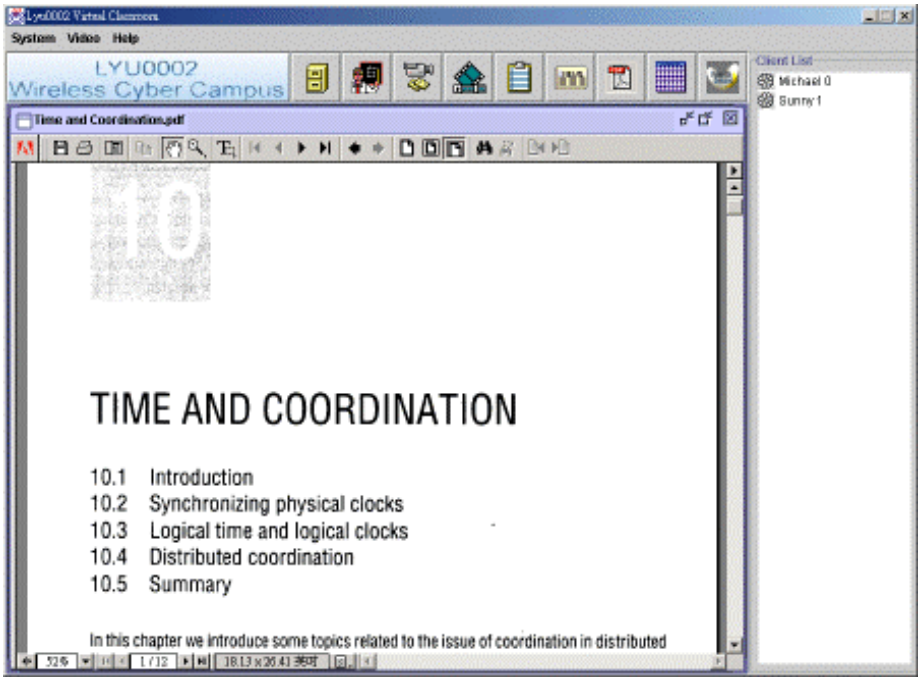


figure 5.12 PDF file reader

Introduction

Lecture notes and reference materials are usually stored as (PDF) format. In order to help students opening the notes to revise, we have embedded Acrobat Reader into our system. Students can read the materials with the Reader of our system to have revision on the materials.

Using the Utilities

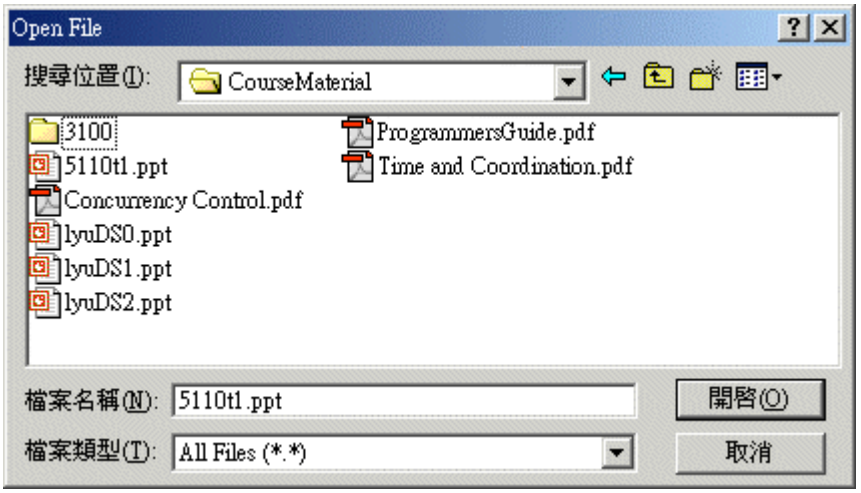


figure 5.13 Dialog box to choose the open file

By clicking the PDF reader icon, the choose file dialog box (fig. 5.13) will pop up for user to choose which PDF file he wants to open.

After choosing the PDF file, Acrobat reader will be opened and the required file is shown in it.

After downloading the PDF file from the file server, the PDF reader will be opened automatically to show the documents.

5.2.4 Preview/Capture web camera 

Introduction

In Cyber Campus, all the lectures are captured and uploaded to the server. Students can attend lectures at anytime by downloading the lecture video or watch through streaming of the video. In order to archive this purpose, our system supports video capturing. So that instructor can use our system to capture the lecture for student reference later.

Using the Utilities

They are two functions in this icon.

Start preview

A preview window will be opened in the main window. This provides a way for instructor to set the web camera up before the live lecture starts.

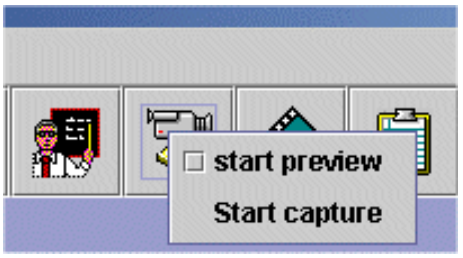


figure 5.14 Preview/Capture menu

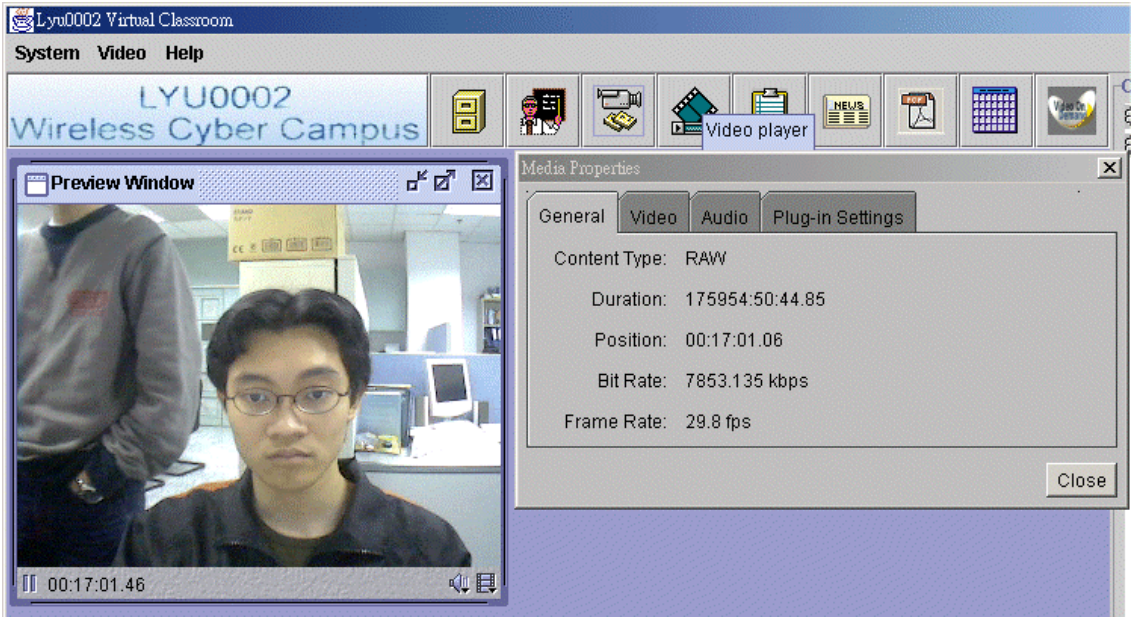


figure 5.15 Preview window and the media properties

Start capture

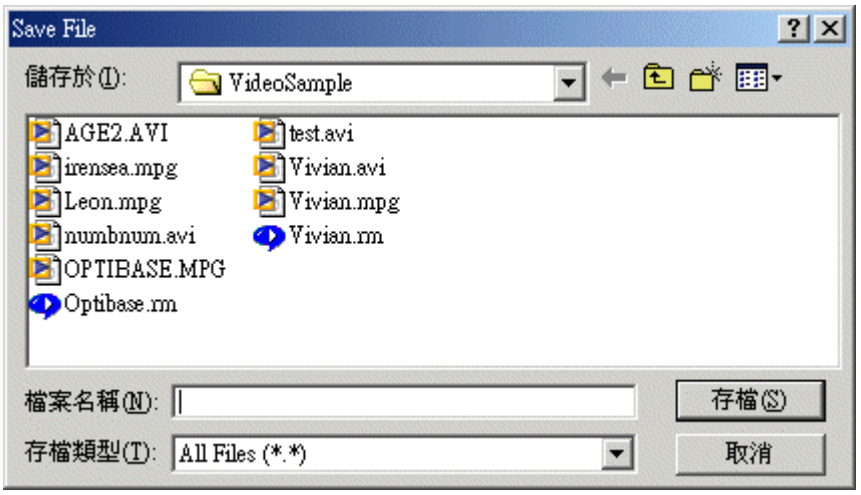


figure 5.16 Dialog box to choose the save file

By clicking the "start capture", the file choose dialog box (fig. 5.16) will be popped up for user to choose where the captured file should be saved.

After choosing the destination, the web camera will start to capture the lecture and store it.

5.2.5 Video playback 

figure 5.17 Video player

Introduction

In Cyber Campus, students can attend the lecture at any time by downloading and playback the lecture video in the server. This video player supports playback of video in AVI and MPEG-I format. Student can use this player to watch the video and have the lesson.

Using the Utilities

By clicking the video playback icon, the choose file dialog box (fig. 5.13) will be popped up for user to choose which video file want to view.

After choosing the file, the player window will be opened and start to play the video.

Students can seek the position for the video to play with. They can pause the video and resume it by clicking the left-bottom button.

5.2.6 Live lecture 

Introduction

A main purpose of Cyber Campus is student can have lessons anywhere. If the lecture is in real-time, students can raise questions whenever they want. In our system, instructor broadcasts the lecture through streaming of the lecture in real-time. Student can have their lessons by receiving the streaming video when the lecture starts. If they have question want to ask, they can send message to the instructor or ask through video conferencing.

Using the Utilities

They are two functions in this icon, start live lecture and start recorded lecture.

A dialog box is shown to enter the details of the target address, port number and time to leave.

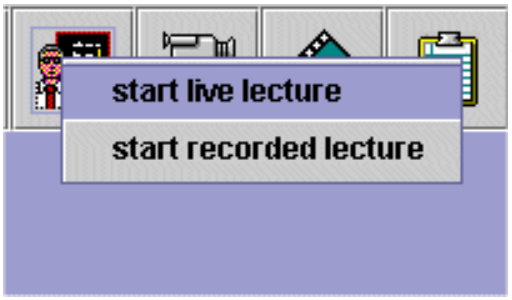


figure 5.18 Live Lecture Menu

For instructor both functions will start to stream the video (live-captured or recorded video) to students

For students both functions will start to receive the stream from instructor.

A dialog box is shown to let users to enter destination IP address, port number and time to leave.

A lecture starts after necessary information entered.

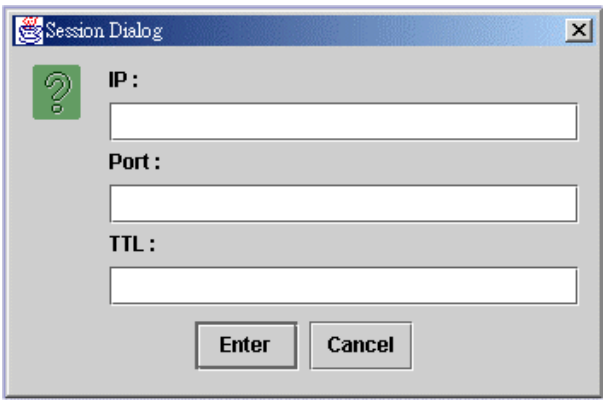


figure 5.19 Session Dialog

5.2.7 Invite video conference

figure 5.20 Video conference window

Introduction

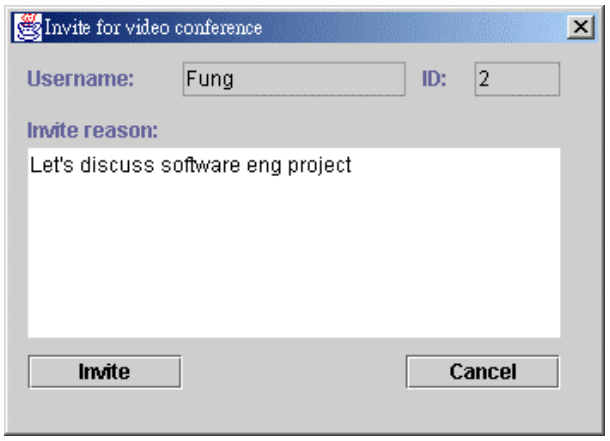
To enhance communication between instructor and students, we would implement video-conferencing. With this function, teachers and students can enjoy closer interactions. Not only students will feel more open and comfortable when they ask questions with teachers, but also teachers can interact with the students more privately and directly.

The interface of the video conferencing is also similar to that of ICQ. So that instructor and students can feel familiar with it and can more easy to use.

Using the Utilities

figure 5.21 Video conference

User can have a video conference with another in the system. User needs to invite other to have a video conference first by right click the name of the user. An invite video conference dialog box (fig 5.22) will be popped up.



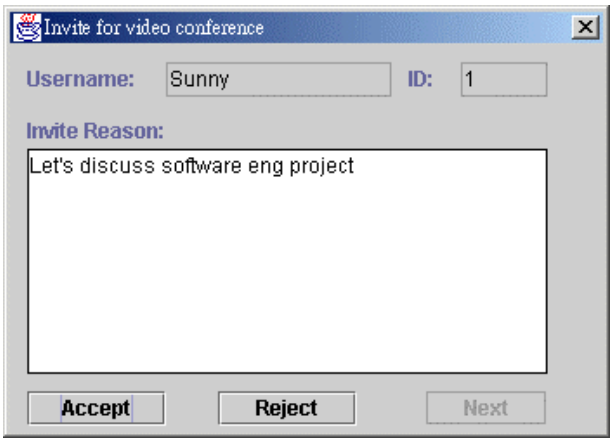


figure 5.22 Invite Video conference

figure 5.23 Reply video conference

After sending the invitation to other. The other side will receive an invitation event like figure 5.24.

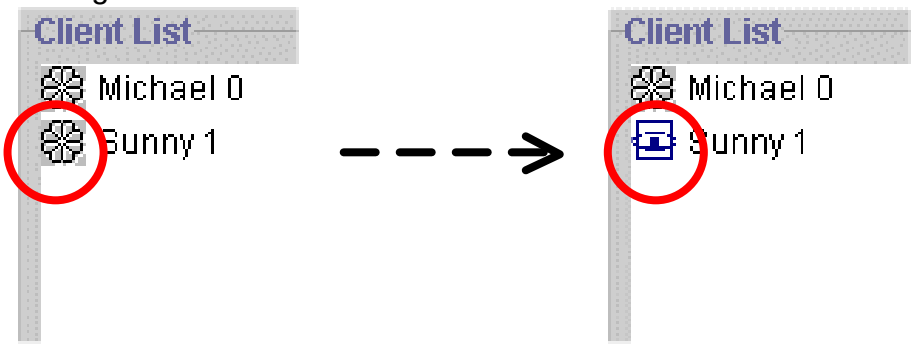


figure5.24 Snapshot of the receive invitation event

When the user being invited double click the inviter' s name, a reply video conference dialog box (fig.5.23) will be popped up. User can choose to accept or to deny the invitation. If user accepts the invitation, both the user and inviter will start the conference.

If one side of the video conference is closed, the other side will be notified and close the video conference.

5.2.8 Video-on-demand 

Introduction

In Cyber Campus, all lecture videos are stored in the server. The server keeps a list of video files stored and shown in the server' s video file dialog. Student can use our system to request for the lecture video stored inside the server. User can filter the type of video they want by searching with key type "lecture", "tutorial", and "ref". Only the relevant video will be shown. Students can then double click the video they want so that they can video the lecture video by streaming.

Using the Utilities

Instructor and student will have different permission to use this function.

If the user is a instructor, (s)he can used both upload video to and stream video from the server. If the user is a student only, (s)he can only stream the video that shared in the system.

figure 5.25 upload video menu

Upload video

Instructor can upload the recorded lecture or any reference video to the server. They can write some description for each video so that student can choose what they want to use.

After uploaded the video, the video can be ***downloaded or streamed*** by the student. All the videos are stored in the server' s hard disk.

Video-on-demand icon

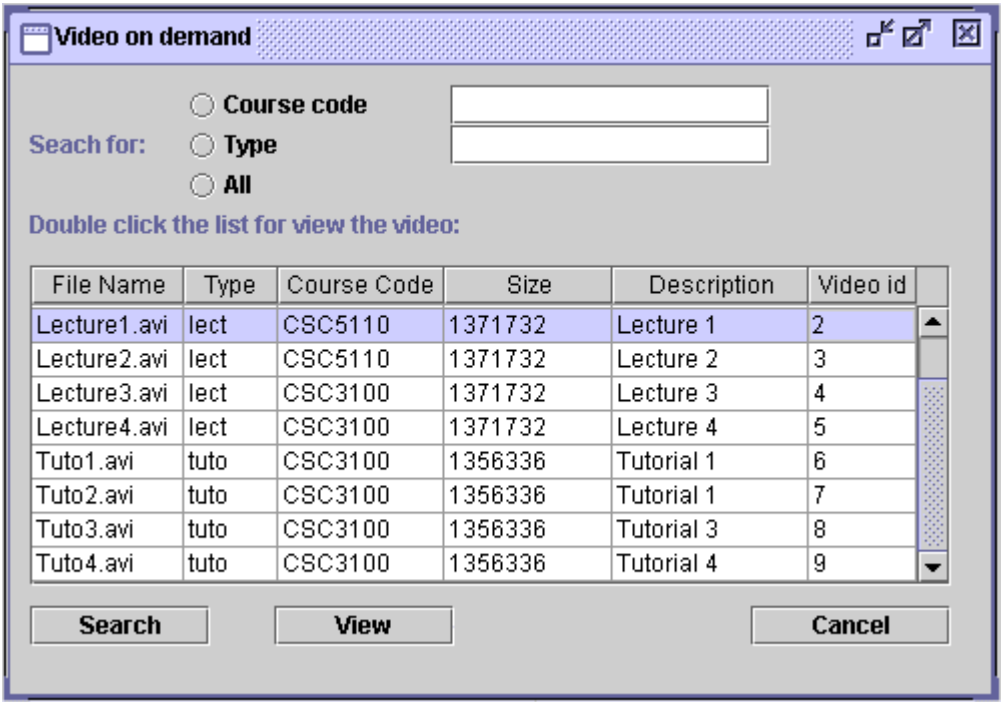


figure 5.26 Video-on-demand dialog

By clicking the Video-on-demand icon, the video-on-demand dialog (fig. 5.26) will be shown in the main window.

In the video-on-demand dialog, there are some features to help user to find the video, which is useful and related to their study.

✍✍ Sorting

User can sort the files to ascending order by clicking the header of that column. After sorting, user can select the video easier.

✍✍ Filtering by course code

By using the filtering feature, user can use the filtering feature to filter the specified course code. The feature is similar to file sharing.

✍✍ Filtering by file type

By using another filtering feature, user can filter the video by specified video type. The feature is similar to file sharing.

User can double click the video to watch it or select the video first and then click the view button in the video-on-demand dialog.

5.2.9 Calendar 

Introduction

A simple calendar used to help student to check for date in a Month.

5.2.10 Notice board 

Introduction

A notice board used to notify students for certain class events such as deadline of certain assignment, examination date.

We have not implement this function yet.

5.2.11 Newsgroup 

Introduction

A newsgroup for students discusses course assignment. Instructor can answer student question in the forum

We have not implement this function yet.

6. Communication between Server and Client

In our system, we have defined several message types for communication between client and server. Below is a table listing the message type and what the type represent.

C->S : Message from Client to Server S->C: Message from Server to Client

Message Type	C -> S	S -> C	Information Stored
INFO		*	ClientID
INFO_S	*		Login Name, Password
UNICAST	*		SourceID, Dest.ID, Message
UNICAST_S		*	SourceID, Dest.ID, Message
LOGIN		*	ClientID
LOGOUT	*	*	ClientID
SERVERDOWN		*	None
FILE_UPLOAD	*		ClientID, Type, FileName, Description, Event ID, Size, CourseCode
FILE_UPLOAD_REPLY		*	FileID, EventID
FILE_UPLOAD_FINISHED		*	FileID, EventID
FILE_UPLOAD_DATA	*		FileID, ClientID
FILE_UPLOAD_INFO		*	FileID, FileType, FileName, Description, Size, CourseCode
FILE_DOWNLOAD	*		FileID, ClientID
FILE_DOWNLOAD_FINISHED		*	FileID, ClientID
INVITE_VIDEO_CONF	*	*	SourceID, Dest.ID, Reason
INVITE_VIDEO_CONF_REPLY	*	*	SourceID, Dest.ID, Response
STOP_VIDEO_CONF	*	*	ClientID
VIDEO_UPLOAD			ClientID, Type, FileName, Description, EventID, Size, CourseCode
VIDEO_UPLOAD_REPLY		*	FileID, EventID
VIDEO_UPLOAD_FINISHED		*	FileID, EventID
VIDEO_UPLOAD_DATA	*		FileID, ClientID
VIDEO_UPLOAD_INFO		*	FileID, FileType, FileName, Description, Size, CourseCode
VIDEO_VOD	*		FileID, ClientID, PortNo

7. Challenges

In our Final Year Project, we faced a great challenge in first semester. We need to learn Visual C++, MFC, Winsock and DirectShow in order to implement our system. However, after we have implemented part of our system, we faced difficulties in implementing DirectShow filters to make video streaming, video-conferencing possible. We have found three ways to solve this problem.

- Use Microsoft's Media Player plug-in to playback the streaming of video in ASF format.
- Use Microsoft's NetMeeting SDK's conference feature.
- Use Java's JMF API

After thorough consideration, we have decided to use Java's JMF API. The fist two methods need external applications, Media Player and NetMeeting. We want to implement the system by ourselves. Therefore we choose JMF as a solution to implement the video streaming and start to rewrite the system in Java.

We need to study Java, Java networking and JMF in order to implement our system in second semester. Java is a completely new language for us. We need to learn it as soon as possible. Luckily, we can finally complete the video streaming and conferencing in our project.

8. Further Extension

We think that our system can be further improved to include more functions. Here are some extensions that can be made.

8.1 Whiteboard

Instructor may sometime need a drawing to express his idea during teaching. For example, in a graph theory lecture, instructor may needs to draw a tree graph to show how to do insertion, deletion and rotation on the tree. Whiteboard allows instructor to show his drawing to students. The drawing of instructor will be displayed on students immediately. This will enhance the efficiency of teaching for instructor to let students have a more understanding on the topics.

8.2 On-line-Quiz

In Cyber Campus, we should include a measurement to test the understanding of students on the topics they learned. On-line-Quiz would be a solution. Instructor can place some quiz on the course for student to test for their understanding. For some multiple-choice type quiz, the result can be shown to students immediate after the quiz. A marking program in the Cyber Campus can do all the markings for instructor. They no longer need to do the marking themselves.

8.3 Global University

Universities all over the world can co-operate together to support such a Cyber Campus so that students can enroll in any courses in the University joined. For example, we can take a programming course in CUHK and a business course in UCLA through the Cyber Campus. We can exchange the resources of the Universities through Internet to make academic exchange more efficient. If this idea makes possible, students no longer need to attend only one University' s course but can choose the courses all over the world.

9. Conclusion

We can simply conclude our work in this year as follows:

- In first semester, we have written application in Visual C++ including chatroom, video player and capturer. However, we face difficulties on further develop our application to include video conferencing.
- In second semester, we have successfully written application in Java including chatting, file sharing, video player, video capturing, video conferencing and video streaming. With these functions, we can simulate a Cyber Campus environment.

As we face some challenge in first semester, we need to rewrite our system from Visual C++ to Java in second semester. Due to limitation on time, some of our functions like newsgroup and notice board have not been implemented. However, we think that it is easy to implement it for our system. Besides, in our project, we have demonstrated a way to use some ActiveX objects like Calendar and Acrobat Reader to be embedded in our Java application. We can further use this technique to enrich our system functionality.

In conclusion, we have developed a system which simulates a Cyber Campus environment. This environment shows a new style of learning and communication for instructor and students. We hope that such environment will be suitable for future education.

10. Acknowledgement

We would like to thank Prof. Michael Lyu, our project supervisor, for giving us valuable advice. He has provided many suggestions and comments to us throughout this project. We are much appreciated by his patience and kindness in advising us.

Moreover, we would like to thanks Edward, Technical Manager of VIEW project, who gives us ideas in our project

11. Reference

1. MSDN Online
   http://msdn.microsoft.com/default.asp
2. Microsoft DirectX 8.0
   http://www.microsoft.com/directx/
3. WinSocket
   http://www.sockets.com/
4. Microsoft NetMeeting Product Home Page
   http://www.microsoft.com/windows/netmeeting/
5. Microsoft NetMeeting Software Developers' Kit
   http://www.microsoft.com/windows/netmeeting/authors/sdk/default.asp
6. Home Page fro Windows Media Player
   http://www.microsoft.com/windows/mediaplayer/en/default.asp
7. Windows Media Format 7 SDK
   http://msdn.microsoft.com/library/psdk/wm_media/wmform/htm/introducingwindowsmediaformat.htm
8. The Source for the Java™ Technology
   http://www.java.sun.com/
9. Java JMF
   http://java.sun.com/products/java-media/jmf/index.html
10. JMF Solutions
    http://java.sun.com/products/java-media/jmf/2.1.1/solutions/index.html
11. Video compression, videoconferencing, and image compression
    http://www-nt.e-technik.uni-erlangen.de/%7Ehartung/links.html
12. Planning and Implementing Wireless LANS
    http://www.networkcomputing.com/netdesign/wlan1.html
13. Mobile Computing related resources on WWW
    http://www.cs.umd.edu/projects/mcml/mcothers.html
14. A Short Tutorial on Wireless LANs and IEEE 802.11
    Daniel L. Lough, T. Keith Blankenship, Kevin J. Krizman
    http://computer.org/students/looking/summer97/ieee802.htm
15. Managing Bandwidth: Deploying QOS In Enterprise Networks, 1/e
    Alistair A. Croll, Ontario, Canada Eric Packman
http://www.phptr.com/ptrbooks/ptr_0130113913.html

16. RSVP Standards Resources
http://www.cs.columbia.edu/%7Ehgs/internet/rsvp.html
17. ISDN Videoconferencing
http://alumni.caltech.edu/%7Edank/isdn/isdn_ai.html - VIDEO
18. The Network Time Protocol (NTP) Distribution
http://www.eecis.udel.edu/%7Entp/ntp_spool/html/index.htm
19. Internet Phone Web Guide
http://www.vocaltec.com/consumer/products/iphone_5/guide/chap4.htm
20. Windows Media Content Development and Deployment
http://msdsn.microsoft.com/workshop/imedia/windowsmedia/default.asp
21. Virtual Classroom
http://www.njit.edu/Virtual_Classroom/
22. The Virtual Classroom and Virtual University at New Jersey Institute of Technology
http://eies.njit.edu/~hiltz/CRProject/sloansum1.html
23. Designing a Virtual Classroom
http://www.njit.edu/Virtual_Classroom/Papers/Design.html
24. Teaching in the Virtual Classroom
http://www.njit.edu/Virtual_Classroom/Papers/Teaching.html
25. Learning Networks: A Field Guide to Teaching and Learning
http://www-mitpress.mit.edu/mitp/recent-books/comp/learning-networks.html
26. The Virtual Classroom: Learning Without Limits Via Computer Network
http://eies.njit.edu/~hiltz/RBooks/ablex1.html
27. J-Integra™ Pure Java-Com Bridge
http://www.linar.com/
28. Mozilla ActiveX Project
http://www.iol.ie/~locka/mozilla/mozilla.htm
29. The Code Project
http://www.codeproject.com/java/
30. Bob Quinn, Dave Shute. Windows sockets network programming. Addison Wesley Pub. Co., c1996.
31. TCP/IP Illustrated Volume 1: The Protocols. (Addison Wesley Professional Computing) by W. Richard Stevens. Hardcover
32. TCP/IP Illustrated, Volume 2: The Implementation (Addison-Wesley Professional Computing) by Gary R. Wright(Contributor), et al. Hardcover

33. TCP/IP Illustrated, Volume 3: TCP for Transactions, Http, Nntp, and the Unix Domain Protocols (Addison-Wesley Professional Computing Series) by W. Richard Stevens. Hardcover
34. Understanding ActiveX and OLE by Chappell, Microsoft Press
35. Java in a NutShell by David Flanagan, O' Reilly & Association Inc.
36. Java Network Programming, 2nd Edition by Elliotte Rusty Harold, O' Reilly & Association Inc.
37. The Java Tutorial Second Edition Object-Oriented Programming for the Internet (Java Series) by Mary Campione, Kathy Walrath.
38. Java Swing, 1st Edition September 1998 by Robert Eckstein, Marc Loy & Dave Wood, O' Reilly & Association Inc.

Appendix A: Progress Report

Month	Description
June	Studying BSD Trying NetMeeting & Internet Phone Studying WinSock
July	BSD UDP Chat-room finished WinChat finished Studying DirectShow
August	Simple Video Player finished Studying MFC
September	MFC Chatting finished WinCap finished
October	Integrated program finished Studying DirectShow Filter Studying Java and Java JMF Java Applet (AVI and MPEG file type supported)
November	JavaPlayer finished JavaCap finished
December	Studying Java Networking Preparing Half-Year report and presentation
January	Text-based Java Client & Server finished Chatting added
February	Studying Java Swing Video Streaming added Live lecture added
March	Graphical User Interface finished Video-Conferencing added Video-On-Demand added File-Sharing added
April	Filtering of Video/File added Acrobat Reader added Calendar added

Appendix B: Statistics of our program

Total number of lines of code:

	Server	Client	Total
Java	3900	7500	11400
Visual C++	1800	5100	6900