

INFORMATION TO USERS

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

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

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

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

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

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

An Online Course Presentation System

Chunbo Yang

A Major Report
In
The Department
Of
Computer Science

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

April 2002

©Chunbo Yang, 2002



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

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

0-612-68488-1

Canada

Abstract

An Online Course Presentation System

Chunbo Yang

This project demonstrates a prototype of online course presentation system. The purpose of this project is to show that that by applying latest multimedia, UI and Web technologies, online course presentation could be more convenient effective and attractive. This Online Course Presentation System is a web-based system used to present the course lectures slides and texts via the Internet. The system is implemented in Java. An applet running in the browser with the powerful multimedia support of Java Media Framework 2.1 (JMF), seamlessly links the applet and Oracle 8i database server using the JDBC technology. The intuitive GUI design makes it looks like a book. Using the system is like reading books. You can randomly choose to any course (book), any chapter or any topic to study. The intelligent interactive quiz subsystem is a unique feature of the system.

This project presents the architectural design of the Online Course Presentation System, the design and implementation of client part GUI and server part database. The key to a successful online system is GUI. So a great deal of effort is focused on the usability and usability testing of this system. Currently (2002 spring term) the system is being used by about 30 students in the Graduate Diploma Program. Finally, future work related to this system is presented.

Acknowledgments

First and foremost, I would like to take this opportunity to express my sincere gratitude to my supervisor, Dr. T. Radhakrishnan, for his agreement with my initial motivation and proposal. Subsequently his enthusiastic support and valuable guidance given to me throughout the project provided me huge support. Without his kind support, nothing would have been made possible.

I wish to thank my friend, Lianggang Yu, Fang Wang, Yiqun Shi, Jing Li and my college Xiaochun Luo, Qiang Jia for their full support. They took time to test the software usability of this project with patience and gave me lots of useful feedback.

I also want thank to all my family members for encouraging me to pursue graduate study and for their moral support during this project. Without their love and support, I could not finish it smoothly.

Finally, I would like to dedicate this project to my wife, Hui Ying, for her full support, great patience and encouragement.

Table of Contents

List of Figures.....	viii
List of Tables	ix
List of Abbreviations	x
1. Introductions and Project Overview	1
1.1 Overview of eLearning	1
1.2 Online Course Presentation System.....	3
1.3 Organization of the Report.....	4
2. Functional Requirement Analysis	5
2.1 Use Cases	5
3. System and UI Design	8
3.1 System Design	8
3.1.1 <i>Web-based Client/Server Computing</i>	8
3.1.2 <i>System Architecture</i>	10
1) Web Subsystem.....	10
2) Presentation Subsystem	11
3) Quiz Subsystem	11
4) Database Subsystem.....	11
3.2 UI Design	12
3.2.1 <i>Overview of Java</i>	12
1) What is Java Applet	13
2) Overview of Swing	16
3.2.2 <i>GUI Design</i>	18
4. Implementation	21
4.1 Web Subsystem.....	21
4.1.1 <i>HTTP & HTML</i>	21
4.1.2 <i>Static vs. Dynamic Web page</i>	22

4.1.3	Implementation	22
4.2	Presentation Subsystem	24
4.2.1	Multimedia Support in Java.....	24
4.2.2	Implementation	24
4.3	Quiz Subsystem	27
4.4	Database Subsystem.....	31
4.4.1	Database & Java.....	31
1)	What is JDBC ?.....	31
2)	JDBC support in Oracle 8i.....	31
4.4.2	Implementation	32
5	Usability Test and Result.....	36
5.1	What's Usability	36
5.1.1	Utility and Usability.....	36
5.1.2	Usability Characteristics	36
5.2	Usability Testing.....	38
5.2.1	Usability Test Design.....	38
5.2.2	Usability Test Result	39
5.2.3	Improvement on UI based on usability testing	42
6	Similar Systems	44
6.1	Features of Course Management System.....	44
6.1.1	Student Functions.....	44
6.1.2	Author Functions	46
6.1.3	Instructor Functions.....	47
6.1.4	Administrator Functions	48
6.2	Comparison of Course Management Systems	48
6.2.1	Comparison of Student Functions.....	50
6.2.2	Comparison of Author Functions.....	52
6.2.3	Comparison of Instructor Functions.....	53
6.2.4	Comparison of Administrator Functions	54
6.3	Features in Online Course Presentation System.....	54

7. Summary and Future Work	56
7.1 Summary	56
7.2 Future Work	57
Reference	59

List of Figures

Figure 2-1: Use Cases	7
Figure 3-1: Architecture.....	10
Figure 3-2: Java Component Structure	13
Figure 3-3: Inheritance Hierarchy of the Applet Class.....	14
Figure 3-4: Life Cycle of Applet	15
Figure 3-5: Common Superclasses of Many of the Swing Components.....	17
Figure 3-6: Overview of the GUI Design	19
Figure 4-1: HTML Page Generated by CGI Program	23
Figure 4-2: Clicking on Topic Node.....	25
Figure 4-3: Clicking on Chapter Node.....	26
Figure 4-4: Clicking on Quiz Node	26
Figure 4-5: Classes Diagram.....	27
Figure 4-6: Quiz.....	28
Figure 4-7: Instant Respond on User's Choice.....	28
Figure 4-8: Show Related Slide.....	29
Figure 4-9: Show Quiz Score.....	30
Figure 4-10: Review on Quiz.....	30
Figure 4-11: Driver-Database Architecture	32
Figure 5-1: Usability Testing Instructions	39
Figure 5-2: Improved UI.....	43
Figure 5-3: Improved UI.....	43

List of Tables

Table 6-1: Access to Course Material.....	50
Table 6-2: Private Space and Customization	51
Table 6-3: Communicate with Others.....	51
Table 6-4: Production of Course Materials.....	52
Table 6-5: Module Management.....	52
Table 6-6: Quiz Features.....	53
Table 6-7: Instructor Functions.....	53
Table 6-8: Features of Online Course Presentation System	54

List of Abbreviations

ASP :	Active Server Page
CGI:	Common Gateway Interface
CMS:	Course Management Systems
DHTML:	Dynamic HTML
HTML:	Hypertext Markup Language
LMS:	Learning Management Systems
LCMS:	Learning Content Management Systems
JDBC:	Java Database Connectivity
JSP :	Java Server Page
JVM:	Java Virtual Machine
SQL:	Structure Query Language

1. Introductions and Project Overview

1.1 Overview of eLearning

With the rapid development of today's Internet technologies, many aspects of our life are impacted by this electronic information revolution. People are talking about E-mail, E-Commerce, E-Trading, E-Magazine, etc. Now, E-Commerce is one of the hottest fields of Internet application. "The next big killer application for the Internet is going to be education. Education over the Internet is going to be so big it is going to make e-mail look like a rounding error. " --John Chambers, CEO, Cisco. IDC (International Data Corporation) estimates that the U.S. corporate market for eLearning will exceed \$7 billion by 2002, representing a compound annual growth rate of 98% from 1997 to 2002.

What is eLearning? eLearning is Internet-enabled learning. Components can include content delivery in multiple formats, management of the learning experience, and a networked community of learners, content developers and experts. eLearning provides faster learning at reduced costs, increased access to learning, and clear accountability for all participants in the learning process.

An eLearning solution includes three aspects, i.e.: *contents*, *tools* and *services*. Let's look into *tools*, the topic we're interested in. *Tools* here mean software tools. There're several functional categories of tools:

- *Authoring Tools* - Essentially, multimedia tools used to create content.
- *Course Management Systems (CMS)* - a software product that generally incorporates an educational delivery system for a topic (typically called a "Course" in traditional

academic educational institutions and most often facilitated via asynchronous-delivery of materials) and provides integrated tools for measuring outcomes and reporting progress for individual or groups of students. CMS systems are currently most popular for educational institutions that are inclined to adapt course based online deliveries.

- *Educational Delivery Systems* - a software product that facilitates the delivery of content or interaction by the web, but is not necessarily intended to comprehensively measure performance over time or handle course administrative tasks.
- *Learning Management Systems (LMS)* - a software product that provides learners an integrated view of all their active coursework and assignments in a "syllabus" that spans multiple courses and that provides comprehensive assessment and goals tracking spanning multiple courses. Within an LMS, learners can select courses, receive content, complete exercises, quizzes, and communicate with instructors and other learners. Instructors, administrators and managers monitor learner's participation through records contained in an LMS.
- *Learning Content Management Systems (LCMS)* - A relatively new category with basic features of a standard Learning Management System (LMS) described above, and enhanced with strong, integrated authoring tools and the capabilities of a knowledge management system.

Due to the limitation of resources, the project *Online Course Presentation System* will just focus on a subset of *Course Management Systems (CMS)*, which is course presentation and quiz.

1.2 Online Course Presentation System

Course presentation and evaluation are the most important parts of all the eLearning systems. The motivation of this *Online Course Presentation System* project is that the student could access the course contents via a web-based system from anywhere at anytime. The system should be easy to learn, easy to use, and the course contents should be efficiently delivered to end users.

The *Online Course Presentation System* project involved web-based client/server computing model. Client side program is developed by using Java applet and Java Swing technologies. Dynamic HTML pages are generated by CGI programs to host the client side applet classes. JMF 2.1 is used to present multimedia contents. JDBC works as interface between Java applet and Oracle8i database server.

Online Course Presentation System has the following features:

1. *Cross Platform*: Client side program is based on 100% pure Java technologies. It could run in any Java-enabled browser.
2. *No installation on client side*: All the HTML pages and Java classes are dynamically loaded from server side. No special installation is needed for a client, which is a big bottleneck for windows based applications.
3. *Heterogeneous Databases*: It can access any JDBC-supported database, such as Oracle, MS SQL Server, Sybase, DB2, etc.

4. *Intelligent link*: Every quiz question is linked to the presentation material (power point slide etc.), which contains related knowledge. This can be used for more infrencing in future than what is implemented now.
5. *Multi audio file format*: All JMF supported audio files could be playback without any modification to the system.

1.3 Organization of the Report

This report describes the architecture, design and implementation of the *Online Course Presentation System*. Usability test and results are also presented. Functional Requirement Analysis is presented in Chapter 2. In this chapter, “use cases” are listed as the analysis result. The System Design and UI Design are described in Chapter 3. System Architecture is a sub topic in System Design. In Chapter 4, the details of four subsystems’s implementation are shown. Chapter 5 presents usability test design and test results. Four similar systems are examined in Chapter 6. They’re Blackboard, LearningSpace, TopClass and WebCT. Chapter 7 provides the summary and future work.

2. Functional Requirement Analysis

Online Course Presentation System is expected to be used as a supplement to classroom-based presentation rather than a complete substitution. This project will achieve the following functionalities in this sense:

- 1) An online tutorial, which allows students to view any of the “modules” within the overall subject of a supported course. It incorporates a random access to the course modules (slides+lectures based tutorial) related to specific topics that a student may like to study or review repeatedly.
- 2) An online quiz, giving students chance to test how well they have learnt the subject matter through well-structured quizzes. The system will give feedback about the students’ answers to the quizzes, help him/her to assess the performance, and show the relevant modules for review, if the student is weak in any of the topics.

2.1 Use Cases

According to the functionalities described above, three kinds of data will be presented to the users: topic materials, lectures (slides+audio), quiz questions and answers. All the data are organized into Courses, Chapters and Topics. All use cases are defined based on how to efficiently present these data to the users and how to give feedback to users to help them go through the study process. The use cases are shown in Figure 2.1.

- Choose Course: the student can choose which course he/she wants to study

The student may take several courses in one academic term. “Switch between courses” is a convenient functionality of a course presentation system. The system should be able to register a student to several courses. Whenever they log into the

system, all the registered courses will be available for presentation. After the student chooses the course, all the related data will be loaded to the system.

- Read Topic Material: the student can read the topic material in text online

Every course consists of several chapters. Every chapter has several topics. For each topic, there's an online text material that explains the knowledge under this topic. The student can randomly access these materials. That means he/she can easily choose any topic.

- Review Slides: the student can review course slides online

Slides are important materials for chapters. Each chapter has a set of slides packaged with audio lectures. The student can review the slides at any time and browse them with navigation buttons.

- Listen to Lecture: the student can control the lecture audio playback or just let it play synchronized with the related slide. Lecture audio playback provides the students a chance to listen the lecture of a real professor that is a partial substitution to classroom-based presentation.

- Take Quiz: the student can take an online quiz for any chapter at anytime.

Quiz is a way to help students to check their study effects. A good quiz helps the student to understand the most important knowledge on that topic.

- Get Feedback: the student can get the feedback from the quiz and review the slides related to quiz questions.

This functionality provides a link between the quiz questions and related knowledge.

It's an effective way to help a student to master the knowledge.

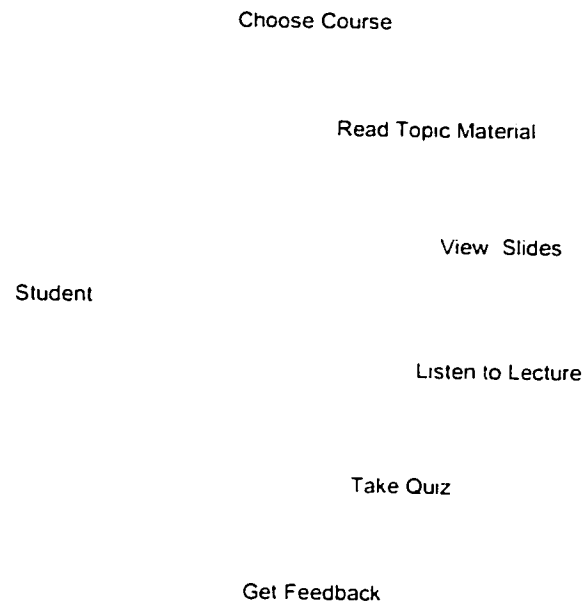


Figure 2-1: Use Cases

3. System and UI Design

3.1 System Design

The Online Course Presentation System is a web-based client/server application. Programs run on both the client's and server's computers. The client part of the program is published from server's computer to client's computer via the web system and runs in a Web Browser environment.

3.1.1 Web-based Client/Server Computing

Client/server computing is a software architecture that involves one or more client processes requesting service from server processes. It is the logical extension of modular programming. The fundamental assumption of Modular programming is to divide a large software into its constituent parts: modules. This creates the possibility for easier development and better maintainability. Client/server computing takes this a step further by executing modules within different memory space. With this architecture, the calling module becomes the "client" (that which requests a service), and the invoked module becomes the "server" (that which provides the service).

The basic characteristics of client/server architectures are:

- 1) Combination of a client or front-end portion that interacts with the user, and a server or back-end portion that interacts with the shared resource. The client process contains solution-specific logic and provides the interface between the user and the rest of the application system. The server process acts as a software engine that manages shared resources such as databases, printers, modems, or high-powered processors.

- 2) The front-end task and back-end task have fundamentally different requirements on computing resources such as processor speeds, memory, disk speeds and capacities, and input/output devices.
- 3) The environment is typically heterogeneous and multivendor. The hardware platform and operating system of client and server are not usually the same. Client and server processes communicate through a well-defined set of standard application program interfaces (API's) and RPC's.
- 4) An important characteristic of client/server systems is scalability. They can be scaled horizontally or vertically. Horizontal scaling means adding or removing client workstations with only a slight performance impact. Vertical scaling means migrating to a larger and faster server machine or multiservers.

The use of Internet technologies for implementing client/server applications brings the client/server system to a brand new Web stage. The explosion of the World Wide Web is due to the world-wide acceptance of a common transport (TCP IP), server standard (HTTP), and markup language (HTML). In its early days of the Web, it was essentially used to publish "read-only" documents. There were limited ways of interacting with the client. With the release of tools such as Java and JavaScript, this limitation is no longer a major issue. The client part of the code can be delivered to client's computer in a real time fashion. The Web-based client/server greatly reduced the problem of managing code on the client. Assuming a standard browser on the desktop, all changes to user interface and functionality can be done by changing code on the HTTP server.

3.1.2 System Architecture

Based on the above investigation on web-based client/server technologies and functional requirement analysis. The system architecture is designed as in the following Figure 3-1. There're four subsystems in this Online Course Presentation System: Web Subsystem, Presentation Subsystem, Quiz Subsystem and Database Subsystem.

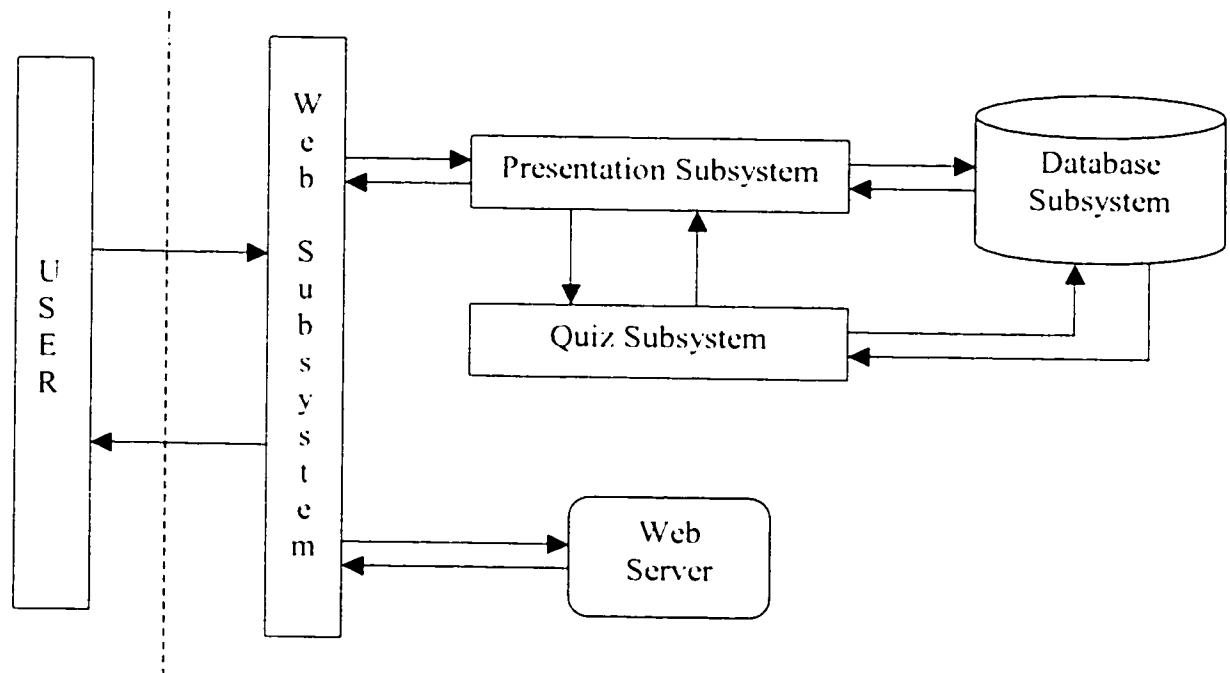


Figure 3-1: Architecture

1) Web Subsystem

This subsystem consists of four parts: Web Browser (client), Web Server, HTML pages and CGI programs. Users launch a web browser in their computers to connect to web server through Internet/Intranet. The HTML page will gather user information and send them as the parameters of CGI program which runs on web server. The CGI program will generate

dynamic HTML web pages according to user's inputs. Within these HTML pages, the Java classes will be loaded from web server to browser and start running on the client's side.

2) Presentation Subsystem

This subsystem has three parts: knowledge tree, slide show and audio. The knowledge tree serves as a table of contents for the courses. Each course has two kinds of nodes representing the two levels of knowledge: chapters' nodes and topics' nodes. Clicking on chapters' nodes will start the process of showing slides as well as playing the lecture related to these slides. Clicking on topics' nodes will trigger showing the knowledge pages (in HTML) of these topics.

3) Quiz Subsystem

This subsystem provides the way to evaluate the effects of studying the knowledge. A user could get feedback immediately or later in a session. The feedback includes answer to the quiz, explanation to the answers and where the user stands in his/her performance, the direct links to the knowledge related to the each questions.

4) Database Subsystem

Oracle 8i is the latest product of Oracle. It supports multimedia data types. The database subsystem has *Course*, *Student*, *KTree* tables which save course information, student information and the knowledge of the courses respectively. JDBC is used to connect to the DBMS.

3.2 UI Design

As a web-based program, the UI plays the most important role of the whole system. UI mediates between the user and the program. It helps users to understand system's functionalities, reflects the system model to them and translates their intentions into appropriate system activity. A good UI helps the user form a model, known as the user's mental model, of how the application works. This model forms the basis for future interactions with the system and enables users to predict system performance. The UI design of Online Course Presentation System follows the real world "book-metaphor". For example, the knowledge is organized into chapters like a book, there's a slide lecture (with audio) at the beginning of every chapter and there's a quiz at the end of every chapter. So the student will begin to study the course from the slide lecture of one chapter then read the materials in every topic, do the quiz, finally, review the knowledge. The Online Course Presentation System is based on Java technology. Its UI is created as Java Applet with Java Swing.

3.2.1 Overview of Java

Java is an object-orient programming language developed by Sun Microsystems Inc. Besides the OO features, Java provides the cross platform ability. As with other high-level computer languages, Java source compiles to low-level machine instructions. In Java, these instructions are known as bytecodes. They are platform-independent instructions, which interact with Java Virtual Machine (JVM). The JVM is a separate program optimized for the specific platform on which the Java bytecodes are executed. The following figure shows how Java can maintain platform independence.

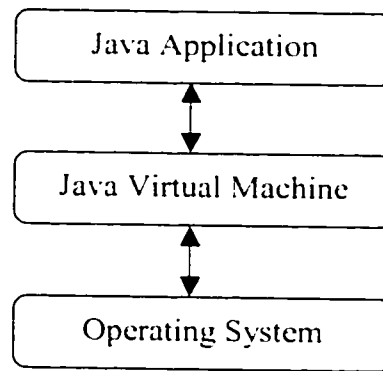


Figure 3-2: Java Component Structure

1) What is Java Applet

An applet is a program written in the Java that can be included in an HTML page, much in the same way an image is included. When you use a Java technology-enabled browser to view a page that contains an applet, the applet's code is transferred to your system and executed by the browser's Java Virtual Machine (JVM).

Unlike a Java application that executes from a command window, an applet is a Java program that runs in the appletviewer (a test utility for applets that is included with the J2SDK) or a World Wide Web browser such as Netscape Communicator or Microsoft Internet Explorer. The appletviewer (or browser) executes an applet when a Hypertext Markup Language (HTML) document containing the applet is opened in the appletviewer (or browser).

Every applet is implemented by creating a subclass of the `Applet` class. The figure 3-3 shows the inheritance hierarchy of the `Applet` class. This hierarchy determines much of what an applet can do and how.

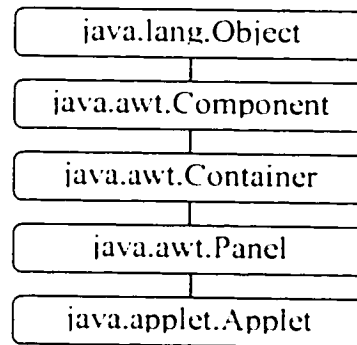


Figure 3-3: Inheritance Hierarchy of the Applet Class

Life cycle of an applet

- Loading the Applet

When an applet is loaded, an instance of the applet's controlling class (an `Applet` subclass) is created; the applet *initializes* itself; the applet *starts* running.

- Returning to the Applet's Page

When the user leaves the page, the applet has the option of *stopping* itself. When the user returns to the page, the applet can *start* itself again. The same sequence occurs when the user iconifies and then reopens the window that contains the applet.

- Reloading the Applet

Some browsers let the user reload applets, which consists of unloading the applet and then loading it again. Before an applet is unloaded, it's given the chance to *stop* itself and then to perform a *final cleanup*, so that the applet can release any resources it

holds. After that, the applet is unloaded and then loaded again, as described in “Loading the Applet” above.

- Quitting the Browser

When the user quits the browser (or whatever application is displaying the applet), the applet has the chance to *stop* itself and do *final cleanup* before the browser exits.

The figure 3.4 summarizes the life cycle of the applet

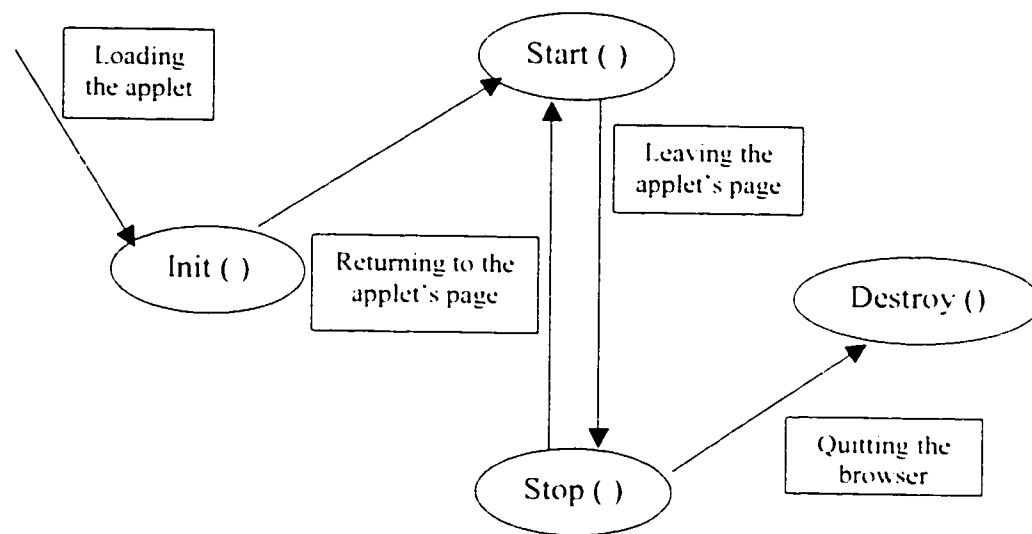


Figure 3-4: Life Cycle of Applet

Applet Capabilities

The `java.applet` package provides an API that gives applets some capabilities that application software does not have. For example, applets can play sounds, which other programs can't do yet. Thus it enhances the web based interactions and dynamic execution capacities of an application software.

The following are what the current browsers and other applet viewers let applets do:

- Applets running within a Web browser can easily cause HTML documents to be displayed.
- Applets can invoke public methods of other applets on the same page.
- Applets can usually make network connections to the host where they came from.
- Applets that are loaded from the local file
- Although most applets stop running once you leave their page, they don't have to.

Security Restrictions

Every browser implements certain security policies to keep applets from compromising the system security. However, the implementation of security policies differs from one browser to another browser. Also, security policies are subject to change. For example, if a browser is developed for use only in trusted environments, then its security policies will likely be much more relaxed than those described here.

Current browsers impose the following restrictions on any applet that is loaded over the network:

- It cannot ordinarily read or write files on the host that's executing it.
- An applet cannot load libraries or define native methods.
- It cannot make network connections except to the host that it came from.
- Windows that an applet brings up look different than windows that an application brings up.
- It cannot read certain system properties
- It cannot start any program on the host that's executing it.

It is such limitations that ensure the network security when an applet is executed.

2) Overview of Swing

The Swing package is part of the Java Foundation Classes (JFC) in the Java platform. The JFC encompasses a group of features to help people build GUIs: Swing provides all the

components from buttons to split panes and tables. The classes that used to create the GUI components are part of the *Swing GUI component* from package *javax.swing*. These are newest GUI components of the Java 2 platform. *Swing components* are written, manipulated and displayed completely in Java (so-called *pure Java components*).

Inheritance Hierarchy of the Classes

The Figure 3-5 shows an inheritance hierarchy of the classes that define attributes and behaviour that are common to most Swing components. Each class is displayed with its fully qualified package name and class name. Much of each GUI component's functionality is derived from these classes.

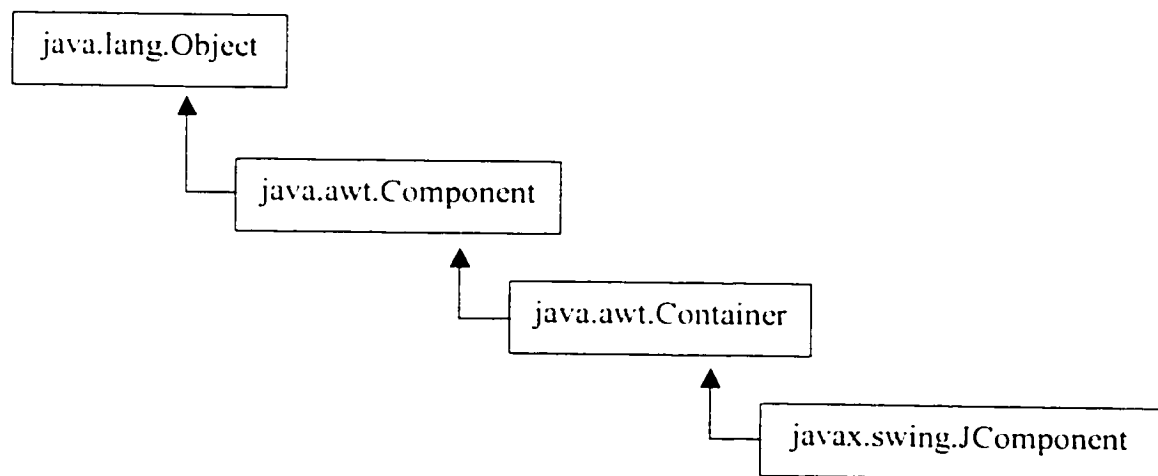


Figure 3-5: Common Superclasses of Many of the Swing Components

Class JComponent is the superclass to most Swing components. This class defines the set of methods that can be applied to an object of any subclass of any subclass of JComponent.

Swing Component's Features

Swing Component that is a subclass of JComponent has the following important features:

- A *pluggable look and feel* that can be used to customize the look and feel when the program executes on different platforms.
- Shortcut keys (called *mnemonics*) for direct access to GUI components through the keyboard.
- Common event handling capabilities for cases where several GUI components initiate the same actions in a program.
- Brief descriptions of a GUI component's purpose (called *tool tips*) that are displayed when the mouse cursor is positioned over the component for a short time.
- Support for assistive technologies such as Braille screen readers for blind people.
- Support for user interface *localization*—customizing the user interface for display in different languages and cultural conventions.

3.2.2 GUI Design

As shown in the Figure 3-6, the GUI of the Online Course Presentation System consists of three parts: the knowledge tree, the knowledge display area and the control panel. The Knowledge Tree serves as a table of contents. The user chooses chapter, topic or quiz using the knowledge tree. The Knowledge Display Area is used to show the slides, the topic material or the quiz questions and result. The Control Panel shows different control components when reading/listening lectures or taking the quiz. It will hide when showing the topic materials. From this single GUI, user can access all the functionalities of Online Course Presentation System. For example, users can randomly access any chapter or topic they want

to study. They can easily control the lecture (slides+audio) playback and quiz process or study the related slides of the quiz questions.

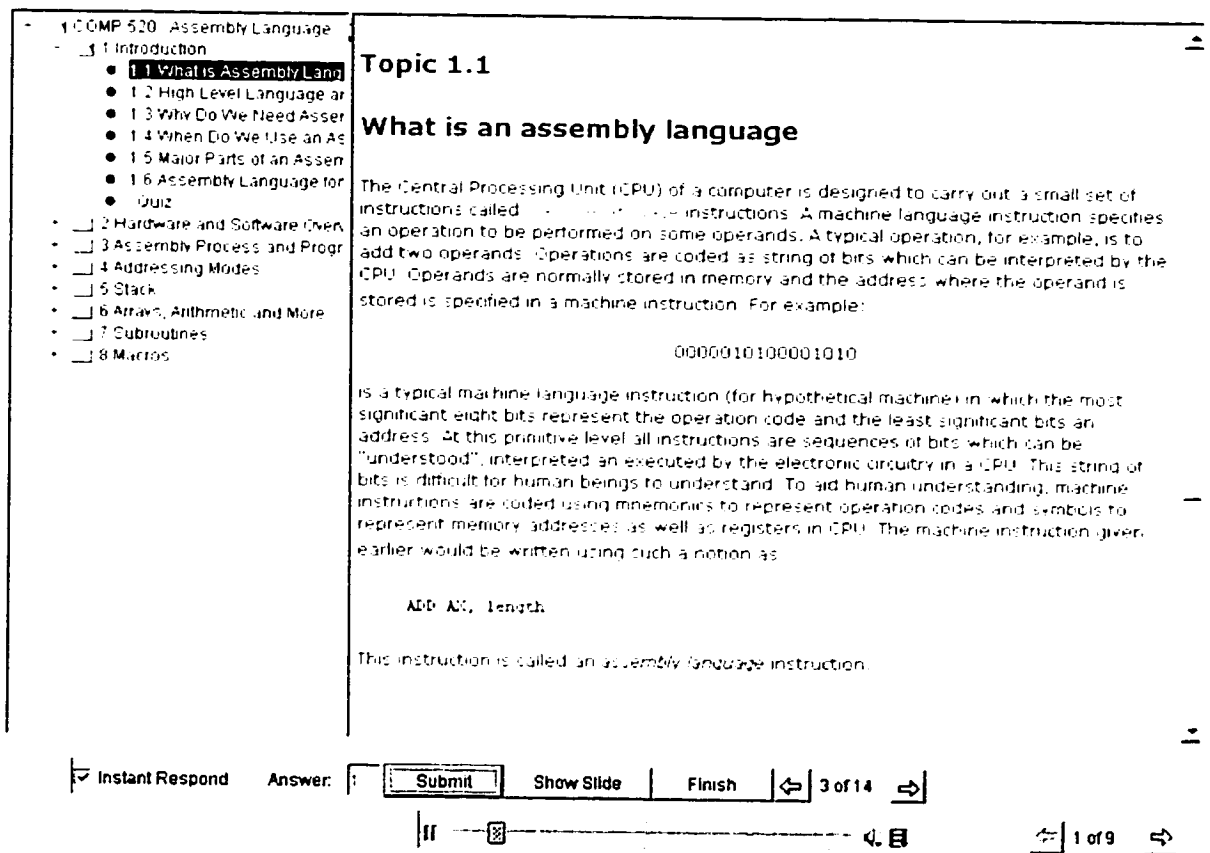


Figure 3-6: Overview of the GUI Design

There are five stages in software life cycle: Analysis, Design, Implementation, Validation and Maintenance. Design is a very important stage in software life cycle. Apart from the obvious criterion that a design should correctly implement a specification, a good design should be understandable and adaptable. To achieve a good design, client/server structure is applied in our architecture design. With this structure, the whole system is divided to two parts: client and server. The client part consists of user related functionalities. For example, GUI components and their event handlers (which are used to handle user's input). The server part consists of shared resources processing operations. Such as: database server and web

server. So the solution-specific logic (which contains in client) and shared resources management (which contains in server) are separated. Further more, the whole system is decomposed to four subsystems: Web Subsystem, Presentation Subsystem, Quiz Subsystem and Database Subsystem according to these functionalities. It provides understandability and adaptability for the whole design. Java technology is used to form a component based GUI design. Based on object-oriented technology and natively applied the MVC modal. Java Swing helps us to design a GUI with more usability and better architecture.

4. Implementation

4.1 Web Subsystem

In Web based systems, CGI programs are used to generate dynamic HTML web pages taking user's input into account. These dynamic HTML pages are delivered from a Web Server to the client's PC via HTTP. The <APPLET> tags in such dynamic pages lead to loading and running of Java Applet classes as client side programs.

4.1.1 HTTP & HTML

HTML - Hypertext is a text that contains links to other texts. The Hypertext Markup Language is the *lingua franca* for publishing hypertext on the World Wide Web. [SK95] It is a non-proprietary format based upon SGML, and can be created and processed using a wide range of tools, from simple plain text editors to sophisticated WYSIWYG (what you see is what you get) authoring tools. It is a collection of platform-independent styles (indicated by markup tags) that define the various components of a World Wide Web document. For example, the matching pair of tags <APPLET> </APPLET> define the Java Applet class in an HTML page.

HTTP –The Hypertext Transfer Protocol is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol that can be used for many tasks beyond its use for hypertext. Examples include name servers and distributed object management systems. A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred.

HTTP has been in use by the World-Wide Web global information initiative since 1990. [RH99]

4.1.2 Static vs. Dynamic Web page

A plain HTML document that the Web daemon retrieves is static, which means it exists in a constant state: a text file that doesn't change. But for web-based client/server applications, we need to create interactive web pages that display dynamically generated contents: such as query results from a database search, customized web pages for different users. With the development of web technologies, more and more ways emerge to generate a dynamic web page. Such as: JSP (Java Server Page), ASP (Active Server Page), DHTML (dynamic HTML). For simplicity, I chose CGI to implement dynamic web pages.

CGI – Common Gateway Interface is a standard for interfacing external applications with information servers, such as HTTP or Web servers. The Web daemon will execute the CGI program on the server side whenever web client invokes it. So, it has the ability to interact with clients in a real time fashion and access the resources on the server side. With this ability the CGI could provide many different services to clients: transmit client's queries to the database engine located at the server, and receive the results back again and display them to the client.

4.1.3 Implementation

There are several ways to implement a CGI program. Perl script, C++ program, etc. In general, the CGI program will take the parameters from web and generate the HTML page according to the parameters. Because a CGI program is running at the server side, it can

access the resources on the server, such as local files, database and other system resources. In “Online Course Presentation System”, the CGI program is a small C++ program. It takes four parameters(tokens):

id: the student ID. such as: 1234567
course: the course number. such as: 520 or 346
from: the knowledge from module. such as: 1,2,3...
to: the knowledge to module. such as: 1,2,3...

(a) Example1: view.exe?course=526&from=2&to=3

means: show page for course 526, module 2 -> 3 show special mark.

(b) Example2: view.exe?id=1234567

menas: show the all courses registered by the student, whose student ID is 1234567.

Let's look at the Example2. First, the CGI program view.exe will take the argument 'id=1234567', parse the tokens in it, get the token values and generate the HTML page as follows:

```
Content-type:text/html
```

```
<HTML>  
<HEAD>  
<TITLE>Web Presentation: 1234567</TITLE>  
</TITLE>  
</HEAD>  
<BODY>  
<CENTER>  
<APPLET CODEBASE="/" CODE="WebPresentation/KTree.class" WIDTH="760" HEIGHT="500"  
ALIGN="BOTTOM">  
<Param Name="ID" Value="1234567">  
</APPLET>  
</CENTER>  
</BODY>  
</HTML>
```

Figure 4-1: HTML Page Generated by CGI Program

As seen from Figure 4-1, it's a simple HTML page which loads an applet class with parameter "ID = 1234567". The *KTree.class* is the 'main' class of Online Course Presentation System.

4.2 Presentation Subsystem

"Presentation Subsystem" is the most important part of the Online Course Presentation System. It controls all the GUI components and the links between them and the database. It also controls the UI to present the quiz information. The Presentation Subsystem takes full advantage of Java's power features to integrate these things: UI, database and multimedia presentation.

4.2.1 Multimedia Support in Java

JMF - The Java Media Framework is an application programming interface (API) for incorporating media data into Java applications and applets. It is developed by Sun Microsystems, Inc., IBM., Silicon Graphics Inc., and Intel Corporation. It is specially designed to take advantage of Java platform features. [JR01] JMF is an optional package of JDK, which can capture, playback, stream and transcode multiple media formats, extends the multimedia capabilities on the Java 2 platform, and gives multimedia developers a powerful toolkit to develop scalable, cross-platform technology.

4.2.2 Implementation

The GUI of the Presentation Subsystem is in the form of a split panel. On the left part of the panel (Figure 4.2) is a tree; we call it K-Tree – knowledge tree. The K-Tree is a three-level

tree with course level, chapter level and topic level. The knowledge of a course is organized into this structure like a book. By clicking on different levels on the K-Tree, different information will be shown in the HTTP window. On the right part of the panel is a HTTP window, which is used to show the slides, topic materials, quiz questions and quiz result. (All of them are in HTML format). At the bottom of the HTTP windows is a small control panel, which is used to control the Slide-Show, lecture audio playback and quiz. All the lecture audios are in MP3 format, and playback with JMF 2.1.1. The control panel also controls the synchronization of showing slides and playing audio.

Example1: Clicking on topic level node will show the topic material.

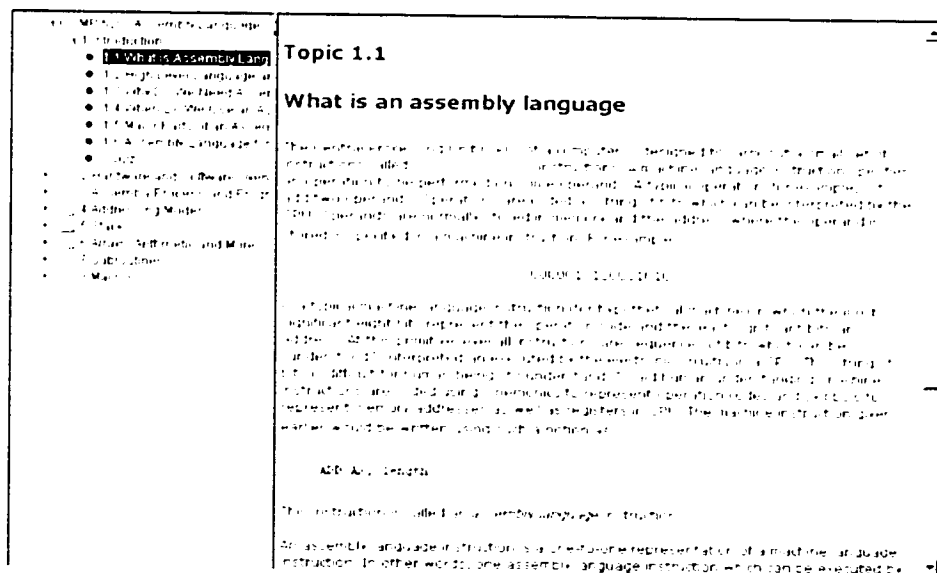


Figure 4-2: Clicking on Topic Node

Example2: Clicking on chapter level node will bring up the slide show and the lecture played back in audio form.

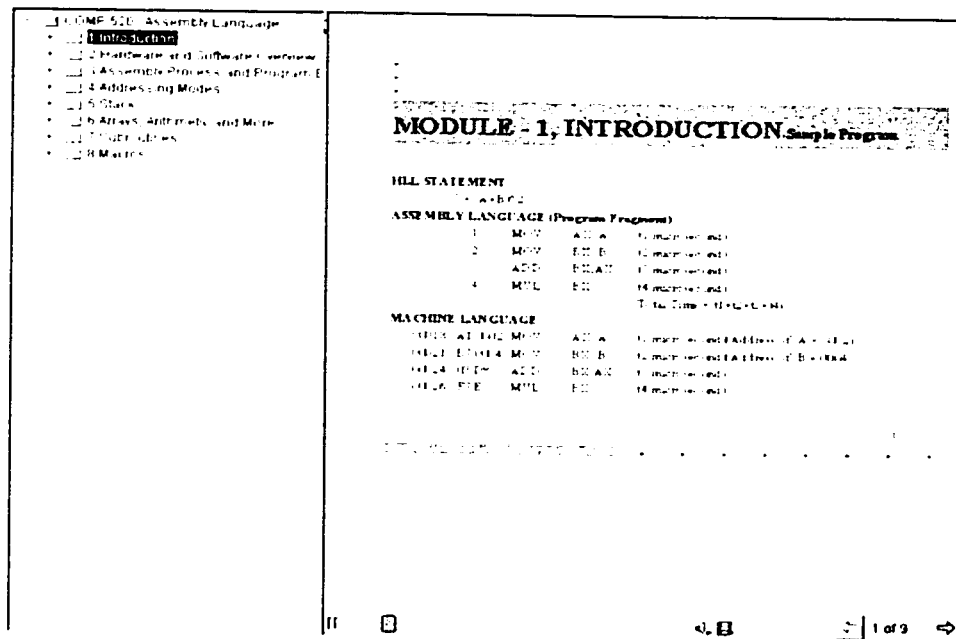


Figure 4-3: Clicking on Chapter Node

Example3: Clicking on quiz node in one chapter will bring up the quiz subsystem.

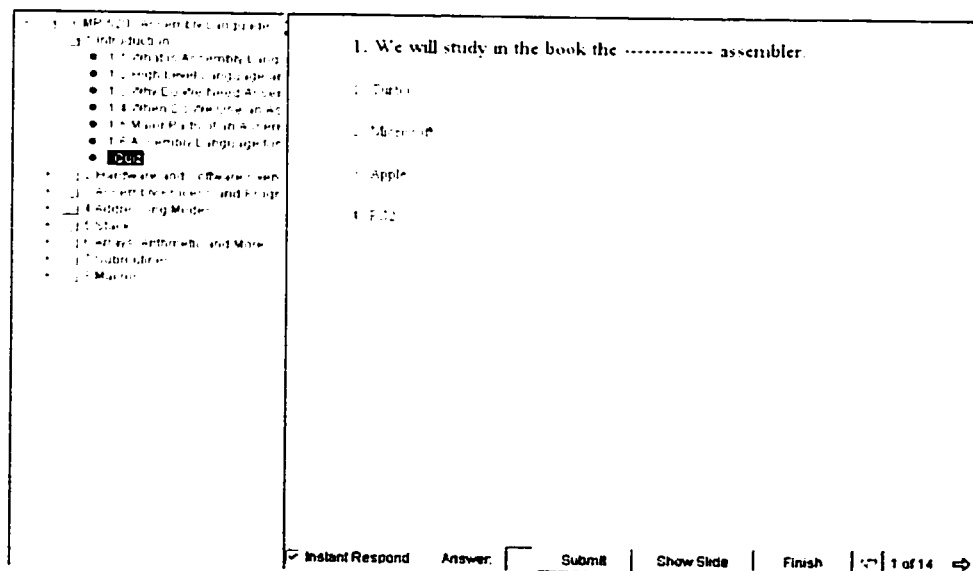


Figure 4-4: Clicking on Quiz Node

The Figure 4-5 is the class diagram of presentation subsystem. From the diagram one can see, the presentation subsystem has followed the MVC model within Java Swing framework.

Class *TreeView*, *HTTPView* and *MControlPanel* implemented VC – View and Control or VC part of MVC and the class *JDBCTreeModel* implemented the M part of MVC model.

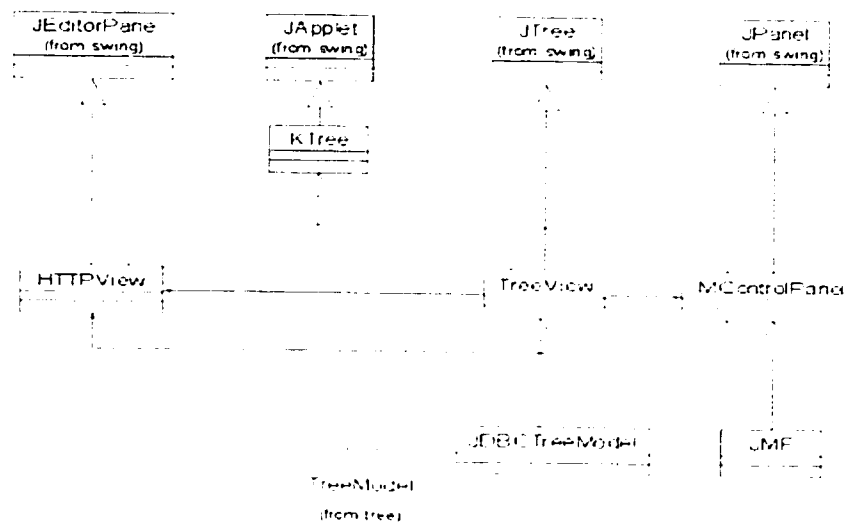


Figure 4-5: Classes Diagram

4.3 Quiz Subsystem

Quiz Subsystem is used as a means for self-evaluation for students to check their depth of understanding. In every chapter's topic node, the last one is always a quiz node, which is connected to the quiz subsystem. The quiz in our case is designed as several multiple choice questions, which relate to the knowledge of the associated chapters. User has to select the right answer from a list of choices. The figure 4-6 is the diagram of quiz subsystem.

The quiz subsystem is controlled by the control panel situated at the bottom part of the GUI. At the left extreme is a check box named *Instant Respond*. This check box is used to control the system to give out the answer immediately after the user clicks the Submit button (see the figure 4-7).

<ul style="list-style-type: none"> COMP 520: Assembly Language 1. Introduction <ul style="list-style-type: none"> 1.1 What is Assembly Language 1.2 High Level Language or 1.3 Why Do We Need Assembler 1.4 When Do We Use an Assembler 1.5 Major Parts of an Assembler 1.6 Assembly Language for Quiz 2. Hardware and Software Issues 3. Assembly Process and Program 4. Addressing Modes 5. Stack 6. Arrays, Arithmetic, and More 7. Subroutines 8. Macros 	<p>1. We will study in the book the ----- assembler.</p> <p>1) Turbo</p> <p>2) Microsoft</p> <p>3) Apple</p> <p>4) POC</p>
<input checked="" type="checkbox"/> Instant Response Answer: <input type="text"/> <input type="button" value="Submit"/> <input type="button" value="Show Slide"/> <input type="button" value="Finish"/> 1 of 14 <input type="button" value="Next"/>	

Figure 4-6: Quiz

<ul style="list-style-type: none"> COMP 520: Assembly Language 1. Introduction <ul style="list-style-type: none"> 1.1 What is Assembly Language 1.2 High Level Language or 1.3 Why Do We Need Assembler 1.4 When Do We Use an Assembler 1.5 Major Parts of an Assembler 1.6 Assembly Language for Quiz 2. Hardware and Software Issues 3. Assembly Process and Program 4. Addressing Modes 5. Stack 6. Arrays, Arithmetic, and More 7. Subroutines 8. Macros 	<p>1. We will study in the book the ----- assembler.</p> <p>1) Turbo</p> <p>2) Microsoft</p> <p>3) Apple</p> <p>4) POC</p>
<input checked="" type="checkbox"/> Instant Response Answer: <input type="text"/> <input type="button" value="Submit"/> <input type="button" value="Show Slide"/> <input type="button" value="Finish"/> 1 of 14 <input type="button" value="Next"/>	

Figure 4-7: Instant Response on User's Choice

In Answer Text field, users could choose their answer. Show Slide button is used to show the slide, which contains knowledge related to this question.

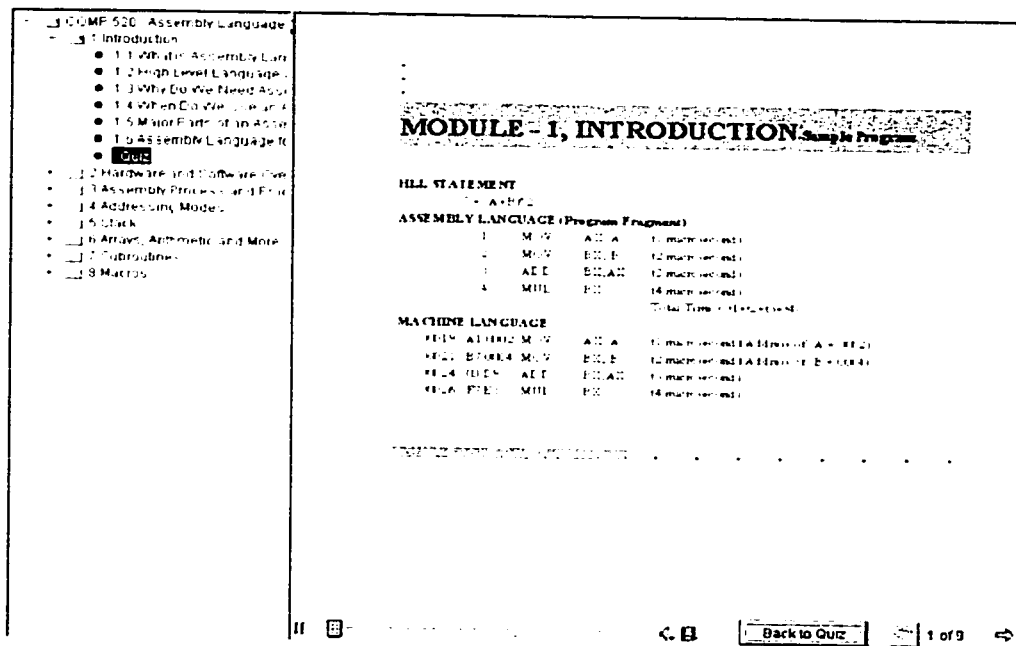


Figure 4-8: Show Related Slide

Clicking on Finish button means the user wants to end the quiz and see the result. If user chooses not Instant Respond then he can click Review button to review his own answers on to the questions and the right answers that will be shown with green or red color indicating the user's answer is correct or not. The two arrow buttons can be used by the user to browse questions with convenience. The page number indicates the current page number and total number of pages. (see next page)

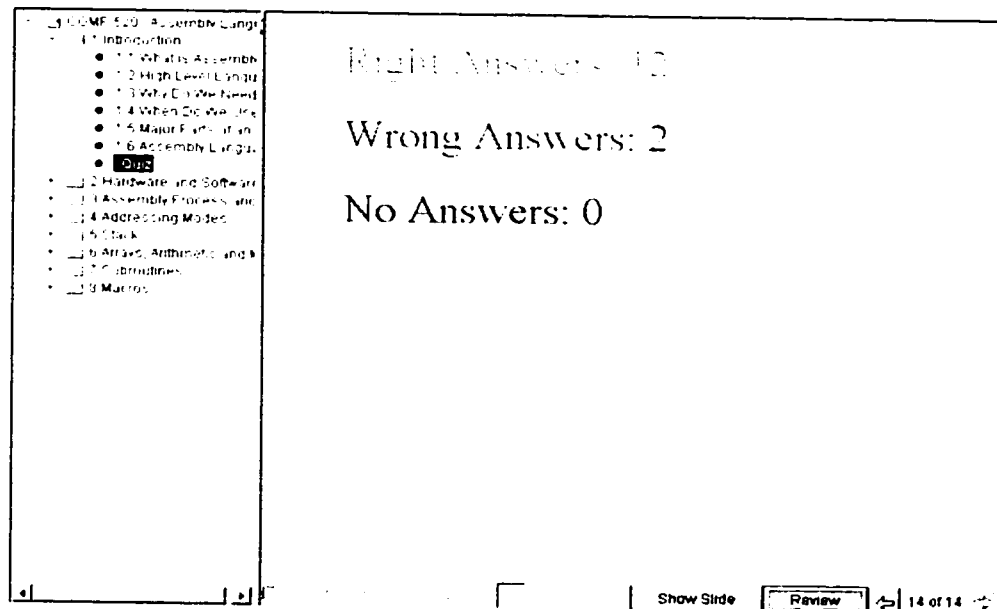


Figure 4-9: Show Quiz Score

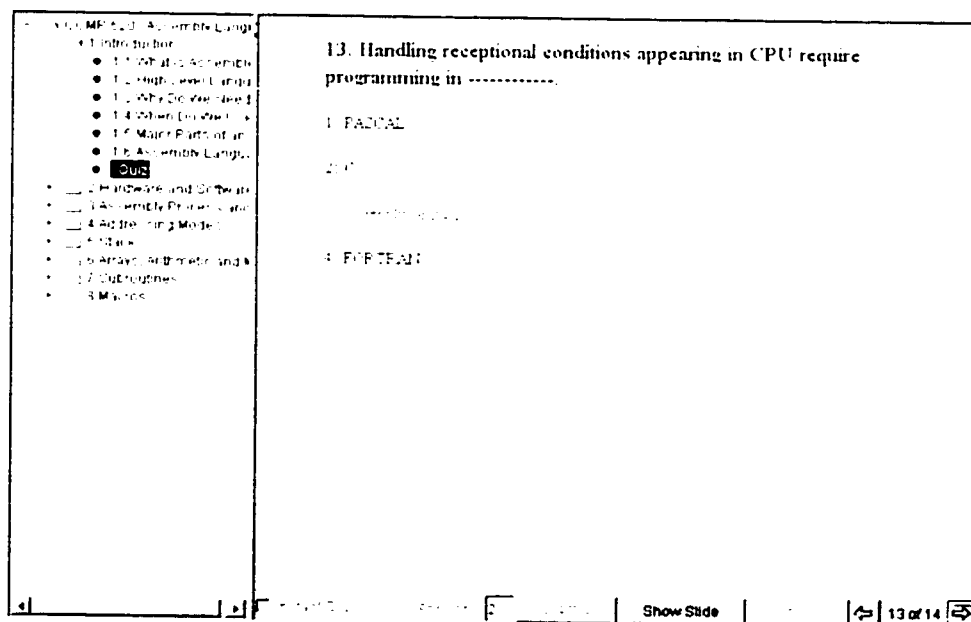


Figure 4-10: Review on Quiz

4.4 Database Subsystem

Oracle 8i is selected as the DBMS of this Online Course Presentation System. It is the latest product of the largest DBMS vendor "Oracle". Oracle 8i has a lot of powerful features to support various applications and various data types that we need. It supports multimedia data type, it provides different JDBC drivers to support Java program, etc.

4.4.1 Database & Java

1) What is JDBC ?

JDBC technology is an API that lets user access virtually any tabular data source from the Java programming language. It provides cross-DBMS connectivity to a wide range of SQL databases, and it also provides access to other tabular data sources, such as spreadsheets or flat files. The JDBC API allows developers to take advantage of the Java platform's "Write Once, Run Anywhere[™]" capabilities for industrial strength, cross-platform applications that require access to enterprise data. JDBC driver is the interface between Java and various DBMS. It converts program (and typically SQL) requests to a form needed by a particular database. With a JDBC technology-enabled driver, a developer can easily connect all corporate data even in a heterogeneous environment. So using the Java programming language in conjunction with JDBC provides a truly portable solution to writing database applications.

2) JDBC support in Oracle 8i

Oracle 8i provides the following JDBC drivers to support different levels' of Java applications.

- (a) *Thin driver*, a 100% Java driver for client-side use without an Oracle installation, particularly with applets
- (b) *OCI drivers* (OCI8 and OCI7) for client-side use with an Oracle client installation
- (c) *server-side Thin driver*, which is functionally the same as the client-side Thin driver, but is for code that runs inside an Oracle server and needs to access a remote server, including middle-tier scenarios
- (d) *server-side internal driver* for code that runs inside the target server (that is, inside the Oracle server that it must access)

The following figure illustrates the driver-database architecture for the JDBC Thin, OCI, and server-side internal drivers. [OJ99]

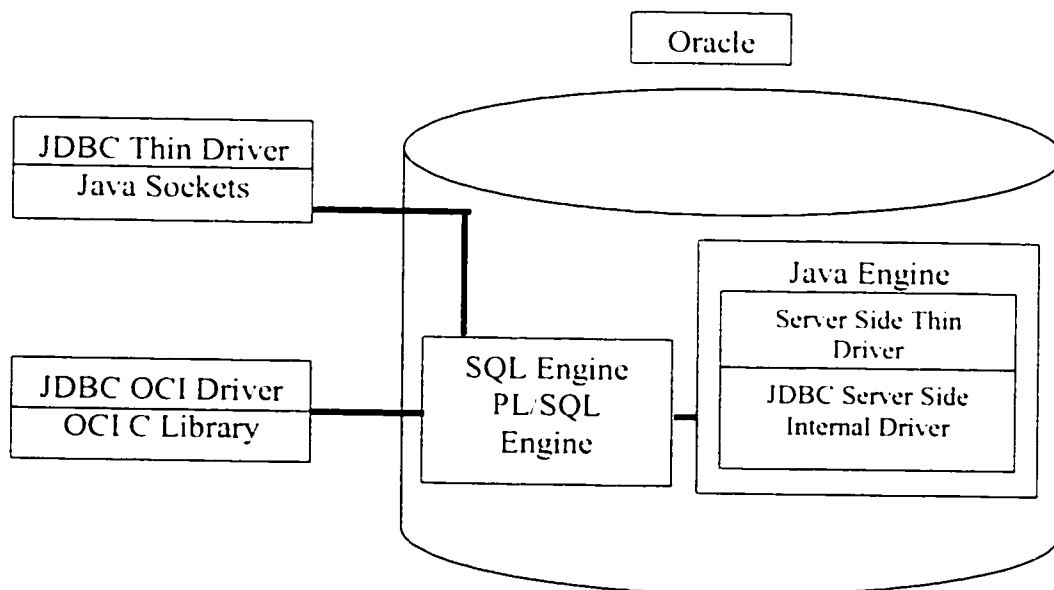


Figure 4-11: Driver-Database Architecture

4.4.2 Implementation

Based on above description about JDBC. I choose Oracle's *Thin Driver* as JDBC driver to access Oracle 8i. With this driver we don't need any Oracle installation on client's PC.

There're three tables in this project database. *Course*, *Student* and *KTree*. Each of them is described below.

Course table has the following fields:

```
TABLE Course (  
    IND          NUMBER    (10),  
    NAME         VARCHAR2  (100),  
    INSTRUCTOR   NUMBER    (10)  
);
```

IND Course Number;

NAME Course Name;

INSTRUCTOR Instructor's ID.

Student table has the following fields:

```
TABLE Course (  
    ID           NUMBER    (10),  
    NAME         VARCHAR2  (50),  
    COURSENO1    NUMBER    (10),  
    COURSENO2    NUMBER    (10),  
    COURSENO3    NUMBER    (10),  
    COURSENO4    NUMBER    (10),  
    COURSENO5    NUMBER    (10)  
);
```

ID Student ID;

NAME Student Name;

COURSENO1 Courses ID 1 that the student takes;

COURSENO2 Courses ID 2 that the student takes;

COURSENO3	Courses ID 3 that the student takes:
COURSENO4	Courses ID 4 that the student takes:
COURSENO5	Courses ID 5 that the student takes.

KTree table has the following fields:

```
TABLE KTree (
    IND          NUMBER    (10)  NOT NULL,
    COURSENO     NUMBER    (10),
    NODEID       VARCHAR2  (10),
    NODETYPE     VARCHAR2  (2),
    NODENAME     VARCHAR2  (100),
    ARTICLE      BFILE,
    QUIZ         NESTED TABLE,
    PRESENTATION NESTED TABLE,
);
```

IND	index of the record
COURSENO	the course number
NODEID	the index of the node
NODETYPE	the type of the node
NODENAME	the name of this node
ARTICLE	the article about the topic
QUIZ	the quiz
PRESENTATION	the slides and audio files

QUIZ is also a nested table of *KTree* table. It has the following fields:

```
TABLE QUIZ (
    IND          NUMBER    (10)  NOT NULL,
    QUESTION     VARCHAR2  (500),
    ANS1         VARCHAR2  (50),
    ANS2         VARCHAR2  (50),
);
```

ANS3	VARCHAR2 (50).
ANS4	VARCHAR2 (50).
ANS	VARCHAR2 (10).
SLIDE	VARCHAR2 (50)

);

IND the index of record

QUESTION the question of the quiz

ANS1 answer 1 to the question

ANS2 answer 2 to the question

ANS3 answer 3 to the question

ANS4 answer 4 to the question

ANS the correct answer

SLIDE the slide contains the related knowledge to this question

PRESENTATION is also a nested table of *KTree* table. It has the following fields:

```
TABLE PRESENTATION (
  IND          NUMBER    (10)  NOT NULL,
  AUDIO        VARCHAR2  (100),
  SLIDE        VARCHAR2  (100)
);
```

IND index of slide and audio

AUDIO the path of audio file

SLIDE the path of slide file

5 Usability Test and Result

5.1 What's Usability

In the interpretation of Usability, we focus on the phrase, "... convenient and practicable for use ..." [WD83]. Usability is the measure of the quality of the user experience when interacting with a product or system -- whether a Web site, a traditional software application, mobile technology like the cell phone, or any other device the user can operate in some way or another. It is also a measure of a product's ability to facilitate completion of users' intended goals, whether their goal is to complete a transaction or process, to find product information, or access a customer service. Key components of creating a usable product are ensuring consistency in the use of elements and navigation facilities, the presence of a clearly defined process and designing the product around customers' goals. A usable product helps to provide a positive customer experience.

5.1.1 Utility and Usability

The **usefulness** of a system is determined by two components:

Utility: Does the system do anything that people care about? If the system does something irrelevant or if it doesn't solve the main problem, then it does not matter whether it is easy to use: it will be a poor system for that user.

Usability: Can the user use the system and can he or she do so effectively? Even if the system does exactly the right thing in theory, it will still be a poor system if the user cannot "figure out" how to get it to work.

5.1.2 Usability Characteristics

Usability is a combination of factors that affect the user's experience with the product or system. It has five characteristics:

1. *Ease of learning*: How fast can a user who has never seen the user interface before learn it sufficiently well to accomplish a set of basic tasks?
2. *Efficiency of use*: Once an experienced user has learned to use the system, how fast can he or she accomplish tasks?
3. *Memorability*: If a user has used the system at some earlier date, can he or she remember enough to use it more effectively next time that may be in the near or far future? (or does the user have to start all over again learning everything every time)?
4. *Error frequency and severity*: How often do users make errors while using the system, how serious are these errors (burning down a cement plant is worse than getting the wrong player's score on a golf site), and how easy is it to recover from a user error?
5. *Subjective satisfaction*: How much does the user like using the system?

Well-designed systems aim to have all the five characteristics of usability. In an individual case, some of the characteristics are more important than others. For the Web, ease of learning is often the most important usability attribute since users rarely spend enough time on any individual Web site to become expert users. Also, subjective satisfaction is critical since users can go anywhere else on the Web at the click of a mouse. User errors are less critical on most Web sites, though E-commerce sites must take steps to ensure that users order the right products and enter their credit card and shipping address correctly. They must

provide suitable error recovery. Intranet designs must focus on efficiency of use since the users of a corporation frequently use the same user interface and their time is money. Extranets need to focus on reducing user errors since they are often used for important transactions.

5.2 Usability Testing

Usability testing is a generic name used to denote a set of methods employed to inspect or examine usability-related aspect of a user interface. The set of methods can be Heuristic evaluation, Guideline reviews, Consistency inspections, Cognitive walkthroughs, etc. [RJ95] Usability evaluators can be end users with content or task knowledge, an intermediary such as system integrator who will integrate the product into a larger product, or usability specialists.

5.2.1 Usability Test Design

The main goal of usability test is to help us find out the problems in the system and resolve them. The *Online Course Presentation System* presented in the report is a Web-based system and how to improve the user interface to make the system user friendly is very important.

The usability test considered the following aspects:

1. Does the system meet the requirements of learning the topics in a course?
2. Is the system easy to use for new users?
3. How long will the users take to become familiar with the system?
4. After being familiar with the system, is there anything that affects the efficiency of using the system? (subjective qualities)

5. Do users like to use the system? How do they feel about the system?

Some usability test cases are based on evaluators and the evaluators will fill a short questionnaire after pre-determined set of testing. During the usability test we conducted, I played as the usability test observer.

Online Course Presentation System

Please try to accomplish the following tasks:

1. Read the first 4 slides in any chapter and listen to the accompanied audio.
2. Reverse/forward to any slide and show it.
3. Read any topic in any chapter then randomly jump to another topic in this or any other chapter.
4. Take the quiz in any chapter with *instant respond* option, read the related slides during the test, review the whole test after submitting all the answers to quiz questions.
5. Take the quiz in another chapter without *instant respond* option, read the related slides during the test, review the whole test after submitted all the answers.

Figure 5-1: Usability Testing Instructions

5.2.2 Usability Test Result

Four computer science students and two software designers from GUI group of Kontron Communications Inc. were selected as evaluators. None of them used this system before but already took the course we provided in the test (Assembly Language). Three of them are

familiar with the knowledge of the course while others are not. The test was done on individual basis to eliminate the affection between them.

Before the test, an oral introduction to the system was given to evaluators. The questions about the functionalities were answered. Such as can I jump to anywhere as I want to? But all the questions about the GUI details and how to use the system are rejected. Such as how can I stop playing the audio? After the test began, no questions will be answered. The evaluator has to accomplish the task following the instructions. Then they answered the following questions:

- Did you have any problem during the test?
 - Yes, sometimes I just want to see the topics in a chapter. But when I click the chapter node, the system starts to show the slides and plays the audio.
 - Yes, no online help is available. For the online system, this is very important, because it's not easy to get help when you are sitting in front of the computer at home.
 - Yes, when I did the same quiz again, the answers I did in the first time is still there.
 - Yes, the waiting time for loading slide and audio is a little long.
- Which part is the most difficult one (if any)?
 - No, the interface is intuitive.
- Would you like to use this system in the future? Why, or why not?
 - Yes, it's convenient. I can use it to study at home. But it's not enough to replace the real class and some bugs should be fixed.

Based on what the evaluators did, what they said, what problems they seemed to have and their answers to the questionnaire, I list some observations from the usability test.

- At the beginning, all the evaluators don't know where the slides are. But they found it immediately after they clicked on the chapter's node. The evaluators also complained about clicking the chapter's node caused the slides showing and audio playing. So the lecture (slides+audio) should not link to the chapter's node.
- Although the UI is very intuitive, online help is found to be necessary
- Long loading time affects the user satisfaction. Shorten the slides loading time, or at least keep the user engaged.
- Some evaluators tried to click the text of the answer to highlight it as selection. And one of the tester input 5 as answer, which is an invalid input. So let users to input number of answer in text filed give them chance to make error.
- Bugs in program also affect the subjective satisfaction.
 - Bug1: When users choose to do a quiz at second time, the previous results are still there, which should be clean up.
 - Bug2: When users reach the last slide, the 'right arrow' button should be disabled because there's no more slide afterward.

Based on all the above observations, I draw some conclusions by checking the usability characteristics.

- Ease of learning: All the evaluators could use the Online Course Presentation System easily without any training. So the system is easy to learn.
- Efficiency of use: Although there's no shortcut key for experienced users, all the major functionalities can be accessed in one main GUI. It's efficient for both experienced and new users.
- Memorability: Users do not need to memorize any command. And the GUI design follows the 'book' metaphor. So users will be familiar with the GUI at the first time they use the system and do not need to learn how to use it afterward.
- Error Frequency and severity: In seldom situation, user input invalid number of the quiz answer. It doesn't affect the system running and user can input another valid number to replace the invalid one.
- Subjective satisfaction: users like using the system, but minor bugs and slow loading process affect their satisfaction.

5.2.3 Improvement on UI based on usability testing

The usability test shows some good results, but still we need to improve the UI.

- Create a lecture node as the first node of every chapter, clicking on it will start the slide show and audio playback.
- Let the user choose quiz answer instead of input answer number.
- Add a help button, which links to a help page.
- Show the status as "loading ..." on the screen when the slides and audio files are being loaded.



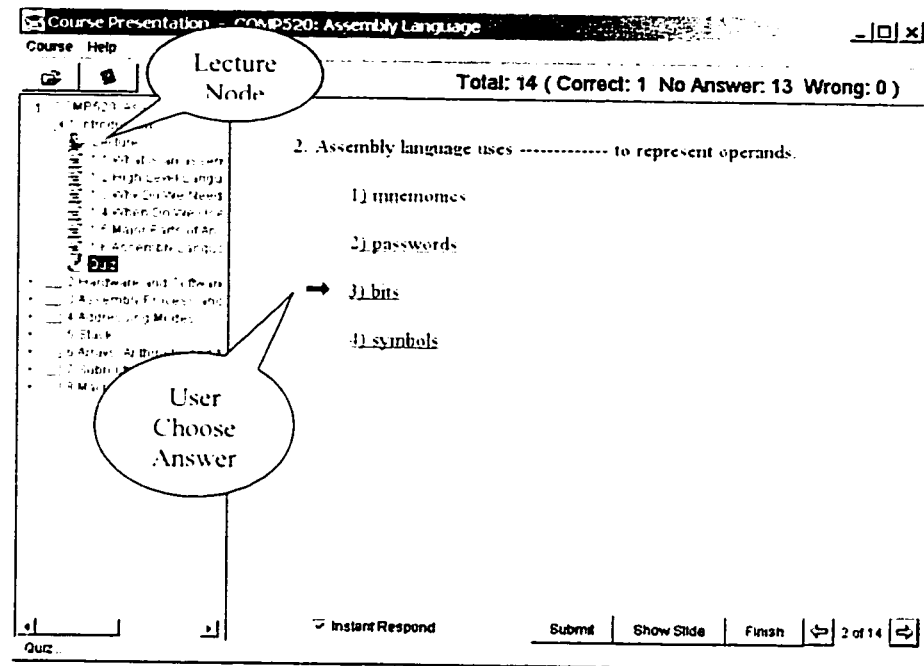


Figure 5-2: Improved UI

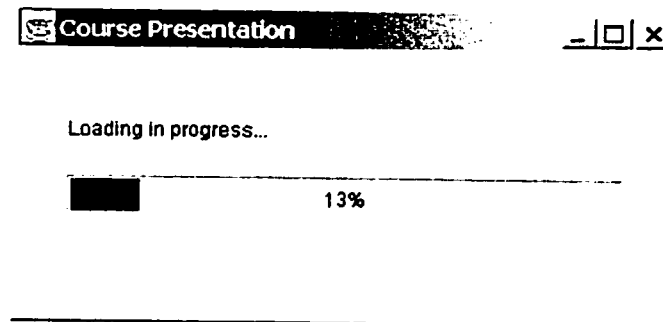


Figure 5-3: Improved UI

6 Similar Systems

This *Online Course Presentation System* is just a subset of Course Management System (CMS). It focuses on course presentation and quiz. For a complete Course Management System, there're some more functions. From user point of view, all the functionalities fall into four categories: Student Functions, Author Functions, Instructor Functions and Administrator Functions. Four CMS systems from the relative are evaluated, based on these four categories in this chapter.

6.1 Features of Course Management System

The followings are some features that are used to evaluate the four CMS systems. These features are listed as four categories as mentioned above.

6.1.1 Student Functions

These functions are used by students who register course(s) in CMS system. With these functions, students can access course materials, access their own private space and customize it, communicate with other student (asynchronous or synchronous) and trace study progress. Some important features are selected to compare between different CMS systems.

- **Access Course Materials:**

- ❑ Friendly UI
- ❑ Keyword Searching
- ❑ Course Download/Offline Working

Entire course or parts of it can be downloaded to a local client.

- ❑ Course Materials Printing

- ❑ CD-ROM Support

Entire course or parts of it can be distributed on a CD-ROM.

- **Private space and Customization**

- ❑ Make Private Note on Course Materials

Student can make note to parts of the course, which are only visible to him/herself.

- ❑ Bookmarks

- ❑ Choice of Learning Sequence

Student can customize the learning sequence, depending on his/her preferences, previous knowledge and educational background.

- ❑ Resume Learning Session

Student can interrupt a learning session at any time. He/She can resume it at the same location where the interruption occurred.

- ❑ Calendar Tool

- ❑ Student Presentation Area

Student can present himself/herself to his/her fellow students. Such as create their own web pages.

- ❑ Username and Password Security

Personal data (private notes, quiz results etc.) is protected at least by a username and a password.

- ❑ Trace Working Progress

Progress tracking includes some facilities for a student to compare his/her learning progress with the learning objectives or with other students.

- **Communicate with Others**

- Asynchronous

- ◆ Email:
 - ◆ Discussion BBS:
 - ◆ Teamwork Tools:

- Synchronous

- ◆ Chat Room
 - ◆ Shared Whiteboard
 - ◆ Audio Conferencing
 - ◆ Video Conferencing

6.1.2 Author Functions

Course author is responsible for creating the course material with authoring tools or third part software and organize them into course structure and publish to the web. Three aspects are considered:

- Production of Course Materials
 - Import/convert existing material
 - Web Interface for Course Development
 - Offline Course Development Interface
 - Index Creation
 - Glossary Support
 - Multiple Authors Support
- Module Management

- ❑ Course Structure Editor/Manager
 - ❑ Curriculum Management
 - ❑ Version Management
 - ❑ Course Resource Management
- Quizzing Features
 - ❑ Quiz Editing Tools
 - ❑ Question Types

Support different question types, such as: single choice, multi-choice, fill in blank, short answer, list matching, etc.
 - ❑ Timed Test Submission

Gives a deadline for a student to submit upload his work test.
 - ❑ Quiz Accessibility Management

Define how the student can access the quiz. Such as: password protected, restricted to be visited only once, restricted to be visible in a specific time slot, etc.
 - ❑ Actions on Test Result

Based on test results the system provides actions like: auto-correct the answers, give a short feedback to the student.

6.1.3 Instructor Functions

In CMS system, the instructors are responsible for tutoring, managing student group and evaluation.

- Tutoring
 - ❑ Asynchronous Tutoring

- Synchronous Tutoring
- Group Management
 - Setup Group
 - Group Files Management
- Evaluation
 - └ Trace Student Progress
 - └ Grade Management
 - └ Statistical Report

6.1.4 Administrator Functions

CMS administrator is responsible for registering student, setting access rights and managing files.

6.2 Comparison of Course Management Systems

Four commercial Course Management Systems are compared in this section. All of them are market-leading products.

- **Blackboard 5.5:** Blackboard 5 is a comprehensive and flexible e-Learning software platform that delivers a course management system (Level One license), and, with a Level Two or Level Three license, a customizable institution-wide portal and online communities. In addition, a Level Three license includes advanced integration tools and APIs to seamlessly integrate Blackboard 5 with existing institution systems. [BB02]
- **LearningSpace 5.0:** The Lotus *LearningSpace* family of solutions enables to create virtual classrooms, build asynchronous courses and create course material and track

students. It's a complete learning management and delivery system that lets you seamlessly integrate the course content, whether you create it yourself or purchase it from Lotus LearningSpace content partners. It has two options:[LS02]

- **Core Module** uses active server page technology and relational database structures to support the delivery and tracking of online self-paced learning content. Self-paced courses can be highly structured with controlled paths and assessments built with the native planning tool, providing flexible learner access designed to suit individual learning needs.
 - **Lotus LearningSpace 5.0 Collaboration Module** combines the self-paced learning capabilities of the core module with the extensive collaborative learning capabilities that enable learners and instructors to work and learn together using discussion databases or real-time, virtual classrooms. This capability makes it possible to bring together groups located throughout the world -- such as a sales force or product team -- and enable them to learn together.
- **TopClass 5.0:** WBT Systems provides with their *TopClass* products a comprehensive training management solution for a wide variety of Web based learning applications. Additionally, instructional design services are offered. To add synchronous communication modes TopClass can be combined with Centra products for virtual classrooms and web conferences. products for virtual classrooms and web conferences. products for virtual classrooms and web conferences.[TC01]

- **WebCT 3.1:** WebCT Inc. is the world's leading provider of eLearning solutions for higher education. It is the preferred partner of more than 2,200 institutions in 77 countries around the world, making it the *de facto* standard in higher education. WebCT 3.1 is an online-based course management system, which is already used by IITS department at Concordia University to provide online courses. WebCT learning tools have many powerful features, such as robust content management capabilities, dynamic learning information management, enhanced learning personalization. [WCT01]

6.2.1 Comparison of Student Functions

All the four systems provide friendly user interface functions to students for accessing course materials. CD-ROM is also support by all the systems. But some of them miss important functions, such as Blackboard and LearningSpace miss keyword searching, which is a very convenient function, and only WebCT provides material printing.

	Blackboard	LearningSpace	TopClass	WebCT
Friendly UI	Y	Y	Y	Y
Keyword Searching	N	N	Y	Y
Course Download	N	Y	Y	N
Offline Working				
Materials Printing	N	N	N	Y
CD-ROM Support	Y	Y	Y	Y

Table 6-1: Access to Course Material

All the private spaces are password protected in these systems. Students' private spaces are allocated based on usernames. In these private spaces the students can trace their own working progress, choose learning sequence, schedule their own calendar or create home page, make private note or set bookmark. WebCT and Blackboard provide most functions in

this category. LearningSpace and TopClass miss some critical ones. Such as TopClass miss bookmark function.

	Blackboard	LearningSpace	TopClass	WebCT
Make Private Note	Y	N	N	Y
Bookmarks	Y	Y	N	Y
Choice of Learning Sequence	Y	Y	N	Y
Resume Learning Session	N	Y	Y	Y
Calendar Tool	Y	N	N	Y
Student Presentation Area	Y	N	Y	Y
Username and Password Security	Y	Y	Y	Y
Trace Working Progress	Y	Y	Y	Y

Table 6-2: Private Space and Customization

All the systems provide asynchronous communications, i.e. email, BBS and teamwork tools. Most of them just provide chat room and shared whiteboard as synchronous communication method. Only LearningSpace provided Audio and Video Conferencing. With the incensement of network bandwidth, more and more eLearning systems add AV conferencing to enable realtime face-to-face teaching. Some of them use their own tools, the others integrated third part products.

	Blackboard	LearningSpace	TopClass	WebCT
Email	Y	Y	Y	Y
Discussion BBS	Y	Y	Y	Y
Teamwork Tools	Y	Y	Y	Y
Chat Room	Y	Y	N	Y
Shared Whiteboard	Y	Y	N	Y
Audio Conferencing	N	Y	N	N
Video Conferencing	N	Y	N	N

Table 6-3: Communicate with Others

6.2.2 Comparison of Author Functions

Powerful authoring tools are essential for creating good course contents. Both integrated authoring tools and third part tools (like HTML editor, image editor) are important. All these systems provide different course development interface and import/convert interface for creating course materials. Multiple authors are support to create the contents for same course.

WebCT has some unique features to create the index and glossary automatically.

	Blackboard	LearningSpace	TopClass	WebCT
Import/convert existing material	Y	Y	Y	Y
Web Interface for Course Development	Y	Y	Y	Y
Offline Course Development Interface	N	Y	Y	Y
Index Creation	N	N	N	Y
Glossary Support	N	N	N	Y
Multiple Authors Support	Y	Y	Y	Y

Table 6-4: Production of Course Materials

All the four systems provide course structure editor manager, curriculum management and resource management. These module management functions help the course author to control the course contents so as to delivery them to student at right moment and in efficient way.

	Blackboard	LearningSpace	TopClass	WebCT
Course Structure Editor/Manager	Y	Y	Y	Y
Curriculum Management	Y	Y	Y	Y
Version Management	Y	Y	N	N
Course Resource Management	Y	Y	Y	Y

Table 6-5: Module Management

Quiz is a very important evaluation method in CMS systems. There're two kinds of quizzes: self-evaluation quiz, teacher-evaluation quiz. Not like the teacher-evaluation quiz, the main purpose of self-evaluation quiz is help the student to master knowledge, so there's no deadline for submission and no access control on it, such as: do quiz only once. Quiz editing tools are provided in all the systems. And all of them have multiple question types, like single choice questions, multiple choices questions. TopClass and WebCT even provide the function of randomly generate questions. Some actions are taken on the test result. Such as: Blackboard and WebCT can automatically correct the answer, give some short feed back and update students' grade book. TopClass can notify the instructor.

	Blackboard	LearningSpace	TopClass	WebCT
Quiz Editing Tools	Y	Y	Y	Y
Multiple Question Types	Y	Y	Y	Y
Timed Test Submission	N	N	Y	Y
Quiz Accessibility Management	Y	N	N	Y
Actions on Test Result	Y	Y	Y	Y

Table 6-6: Quiz Features

6.2.3 Comparison of Instructor Functions

These four CMS systems provide pretty complete tools for tutoring and managing the students.

	Blackboard	LearningSpace	TopClass	WebCT
Asynchronous Tutoring	Y	Y	Y	Y
Synchronous Tutoring	Y	Y	N	Y
Setup Group	Y	Y	Y	Y
Group Files Management	Y	N	N	Y
Trace Student Progress	Y	Y	Y	Y
Grade Management	Y	Y	Y	Y
Statistical Report	Y	Y	Y	Y

Table 6-7: Instructor Functions

6.2.4 Comparison of Administrator Functions

All these systems provide online registration of student account, powerful student files management and access control on different account, such as: student account, group account, instructor account, etc.

6.3 Features in *Online Course Presentation System*

As mentioned in previous sections, our Online Course Presentation System is just a subset of Course Management System. It focuses on course presentation and quiz. So it does not integrate any authoring tools or management tools for student, author, instructor and administrator. It provides only the accessibility to course materials and quiz features. So we checked the features in these two topics.

Features	Online Course Presentation System
Access Course Materials	
Friendly User Interface	The usability result shows it has intuitive UI and easy to use
Keyword Searching	No this function
Course Download/Offline Working	It provides offline version on C'D-ROM.
Course Materials Printing	No printing function.
CD-ROM support	It can read image and audio files from CD-ROM.
Quiz Features	
Quiz Editing Tools	No quiz editing tools
Question Types	Single choice questions
Timed Test Submission	No time limitation for quiz submission
Quiz Accessibility Management	No access control on quiz
Actions on Test Result	It provides instant or later respond choice on quiz answers. It links the quiz answers to related slides.

Table 6-8: Features of Online Course Presentation System

From above table we can note that, the *Online Course Presentation System* provides some good features of accessing course material and quiz, such as offline and CD-ROM support, links between quiz question and related slides. But it also misses some important features. For example, keywords searching, printing material, quiz editing tools, different quiz question types, etc. So some future work has to be done to improve the system.

7. Summary and Future Work

7.1 Summary

This *Online Course Presentation System* is a web-based course presentation and evaluation system. It is intended to be a supplement to classroom-based presentation and evaluation rather than a complete substitution. It helps the students to access the course contents via the Internet from anywhere at anytime. It also provides quizzes for the students to do self-evaluation.

As a web-based system, in our prototype usability is considered as a key aspect from the beginning. To improve the ease of use of the system, the UI design follows 'book' metaphor to help users to create mental model of using the system. Contents and how to organize the contents are vital important to the usability of a web system. So this Online Course Presentation System provides multimedia presentation (slides+audio) feature by using Java Media Framework. It also provides a unique feature of linking the quiz question to the related topics in the slide presentation. Quiz questions are designed to evaluate the effect of students' study. So the knowledge involved in a quiz question is the most important. In this system, when the student is doing or reviewing the quiz, he/she can easily find the related material and review them online by just clicking a button. This feature improved the effectiveness of using the system. The usability test shows good results and potential to improve usability further.

7.2 Future Work

Of course, this project is far from perfect. A lot of future work should be done to improve the system. Based on priority, the following improvements could be done.

- More knowledge links should be added. Such as links between key words in course materials and slides, links between quiz questions and course materials.
- The *Online Course Presentation System* is a two-tier client/server model. It has some problems comparing to three-tier or multi-tier model. A two-tier architecture is where a client talks directly to a server. It is typically used in small environments (less than 50 users). When the server becomes overwhelmed, the system can be ineffective. As we know, a web application may serve hundreds or thousands of users simultaneously. Scalability is crucial to this kind of online system. A three-tier architecture introduces a server (or an "agent") between the client and the server. The role of the agent is manifold. It can provide translation services (as in adapting a legacy application to a client server environment), metering services (as in acting as a transaction monitor to limit the number of simultaneous requests to a given server), or intelligent agent services (as in mapping a request to a number of different servers, collating the results, and returning a single response to the client). [FAQ01] So a major change has to be incorporated to the architecture to improve the scalability and flexibility of the system.
- CGI is a traditional way to generate dynamic HTML page. Some new technologies are more efficient, powerful and portable than CGI. Such as JSP, ASP, PHP, etc.
- MP3 compression technology is used to compress all the audio files in our system. Sometimes, loading takes intolerable amount of time. MP3 is not suitable for live

audio presentation. A/V Multimedia streaming technology is a possible solution to consider in future.

eLearning system provides us cost saving, and geographically wide accessibility. The development of technology is endless. eLearning system are becoming more and more powerful and usable, and supplement the traditional courses.

Reference

- [RJ95] Robert L. Mack, Jakob Nielsen, Usability Inspection Methods: Executive Summary. Readings in Human-Computer Interaction: Toward the Year 2000, Morgan Kaufmann Publishers, Inc. pp.172, pp.176-179,1995.
- [WD83] Webster's Ninth New Collegiate Dictionary, Merrian-Webster Inc. 1983.
- [SK95] R. Steinmetz and K. Nahrstedt, "Multimedia: Computing, communications & applications", Prentice Hall PTR, 1995.
- [RH99] Request for Comments:2616, Hypertext Transfer Protocol -- HTTP 1.1,The Internet Society, pp1,1999
- [JR01] Introduction to Java Media Framework, Sun Microsystem Inc., 2001.
- [OJ99] Overview of the Oracle JDBC Drivers, JDBC Developer's Guide and Reference, Release 2 (8.1.6), pp. 27- 28, 1999.
- [FAQ01] Lloyd Taylor, What is a Three-Tier Architecture? Client Server Frequently Asked Questions, ABSnet Internet Services, 2001.
- [BB02] Corporate Fact Sheet, Blackboard Inc., pp.3, 2002
- [LS02] About LearningSpace, IBM/Lotus Inc., 2002
- [TC01] Topclass 5 Data Sheet, Wbt Systems Inc., pp.1, 2001
- [WCT01] Leveraging Technology to Transform the Educational Experience, A WebCT White Paper, WebCT inc., pp.1-2, 2001.