

University of Nevada, Reno

**Implementation of the Integrated Courseware Manager for
Computer Science**

A professional paper submitted in the partial fulfillment
of the requirements for the degree of
Master of Science
with major in Computer Science

By,
Kishor K. Waikul

Dr. Frederick C. Harris, Jr., advisor

May 2003

Abstract

In this paper we discuss the features of CS.UNR E-Learning system, which is an integrated courseware manager program focused mainly on the evaluation and feedback aspects of computer science courses. It was developed to accelerate the grading process in highly populated courses with frequent quizzes and programming assignments. Additionally, it is an object-oriented system with modular design, which gives developers flexibility to extend the functionality of the existing modules.

Acknowledgement

My deepest appreciation goes out to my advisor, Professor Harris for his extraordinary level of support and patience. I will always remember those countless hours he spent with me in meetings and presentations. His leadership, expertise and advice has been a great motivation for me and has had a key role in making this a unique and rewarding Masters project.

I am very grateful to Professor Looney and Professor Biondi for their valuable time and suggestions and for serving on my committee.

I would also like to thank Professor Harris's wife for her willingness to proofread this paper and for her guiding insights and comments.

On a more personal note, I wish to thank my parents Mr. Kamlakar Waikul and Madhuri Waikul for their support and encouragement throughout my education.

Contents

Abstract	i
Acknowledgement	ii
List of Figures	v
1 Introduction	1
2 Background Literature	2
2.1 Definition of courseware	2
2.2 Courseware development methodology	2
2.3 Standards for courseware development methodology	3
2.4 Computer assisted online assessment and feedback	4
2.5 Existing online courseware systems	6
2.5.1 WebCT	6
2.5.2 Blackboard	9
2.5.3 Centra Symposium	13
2.5.4 eMind	14
2.5.5 Macromedia web learning studio	17
2.6 Motivation for CS.UNR E-Learning	19
2.7 Object-oriented web development	21
2.7.1 ASP and ASP.NET	21
2.7.2 Important aspects of .NET framework	22
2.7.3 Common language runtime	24
2.7.4 Managed, unmanaged and unsafe code – Understanding the difference	25
2.7.5 .NET framework libraries	25
2.7.6 Summary – Benefits of ASP.NET	26
3 CS.UNR E-Learning Features	28
3.1 Instructor’s workspace	28
3.1.1 Course tools	28
A. Add course	28
B. Edit course	30
C. Events manager	31
D. Resource manager	33

E. Class organizer	34
F. Course schema/composition	36
3.1.2 Student management tools	37
A. Show students	37
B. Turn on/off course registration	37
C. Manage grade book	38
3.1.3 Assignment tools	39
A. Add questions	39
B. Edit questions	40
C. Manage files	41
D. Feedback on files	43
E. Preview files	44
3.2 Student's workspace	45
3.2.1 Course tools	45
A. Register for courses	45
3.2.2 Assignment tools	47
A. File uploads	47
4 Conclusion and Future Work	48
4.1 Conclusions	48
4.2 Future work	48
4.2.1 Instructor's workspace	48
A. Create assignments	49
B. Grading daemon	49
C. Email options	49
D. Discussion board	49
E. Course website manager	49
F. Help	50
4.2.2 Student's workspace	50
A. Grade book	50
B. Assignment/quizzes	50
C. Manage files	50
D. Test run assignments	51
E. Answer quiz	51
F. Discussion board	51
G. Help	51
4.2.3 Teaching assistant's workspace	51
4.2.4 Administrator's workspace	52
Bibliography	53

List of Figures

2.1	Key stages in methodology for online feedback and assessment	5
2.2	Designer Map	7
2.3	My WebCT screen	7
2.4	Calendar of events	8
2.5	“My Institution” portal interface	10
2.6	Communication tools	11
2.7	Instructional templates	17
2.8	Quiz framework	18
2.9	Request-response block diagram	26
3.1	Instructor’s workspace and list of courses registered by instructor	29
3.2	Add course form	30
3.3	Edit course form with calendar shown for updating dates	31
3.4	Add events	32
3.5	Show events for the week of selected date	32
3.6	Today’s events	32
3.7	New resource	34
3.8	Show resources	34
3.9	Class organizer	35
3.10	Rearrangement of classes after deletion	35
3.11	Course schema or course composition	36
3.12	Students list from ‘Show students’ option	37
3.13	Course registration switch	37
3.14	Grade formula	38
3.15	Edit grades	38
3.16	(a) Add question: Multiple choice	40
	(b) Add question: Short answer	40
	(c) Add question: True/false	40
3.17	Manage files	42
3.18	Post feedback on solution files	43
3.19	Preview commented files	44
3.20	List of courses open for registration	45
3.21	Course information	45
3.22	File upload for assignments	46

Chapter 1

Introduction

Today's educational landscape is characterized by rising enrollments, changing demographics, and greater demand for anytime/anywhere learning. Expectations by all stakeholders in the e-Education enterprise have never been higher. The use of computers to provide an integrated environment for teaching has received much attention in recent years. Many teaching methods used in the classroom are interactive and can take many forms depending on the desired participation of students. It is difficult and/or time consuming to cover all possibilities in a lecture (or on static media such as handouts and textbook). Integrated courseware has an obvious advantage in presenting a course through asynchronous learning and online evaluation. CS.UNR e-learning is an integrated courseware manager currently focused on the evaluation and feedback aspects of computer science courses. This program was designed to speed up the grading process and provide faster feedback on assignment submissions.

Chapter 2 presents background literature of online courseware development along with a review of some of the most popular online courseware systems. Chapter 2 also discusses the technical features of the ASP.NET platform chosen for development. In Chapter 3 we explain different modules of this integrated courseware program, and in Chapter 4 we cover our conclusions and some ideas for future work.

Chapter 2

Background Literature

2.1 Definition of courseware

The field of courseware engineering, a division of the discipline of educational technology, is relatively young. Thus, there is no consensus on the definition of many terms related to it, such as educational software, courseware, authoring environments and learning environments. However, the definition of courseware presented by Zhiting in [4] is widely acceptable:

“Courseware refers to content-specific instructional software which functions to generate instruction with the support of instructional delivery systems. A courseware product involves five elements: the content and the learning/pedagogical methods as its main components, the learning objectives and the medium as its attributes and the architecture, which organizes the courseware in a way convenient to use.”

2.2 Courseware development methodology

Courseware development is not an easy task. A common illusion about Open and Distance Learning (ODL) is that it is sufficient to take a course syllabus and course material from on-campus teaching and deliver it to students *via* Internet (using WWW,

News, mailing lists, *etc.*) [5]. Courseware development for ODL requires a methodology suitable for the specific subject to be taught, the audience to be addressed, and the specific learning environment.

A courseware development methodology (CDM) discussed in [6] is expected to give answers to all problems encountered in the development process. With regard to the information system development methodologies illustrated in [7], a courseware development methodology can be defined as:

“A collection of procedures, techniques and tools, and documentation aids which will help the courseware developers in their efforts to implement a new courseware. A methodology will consist of phases, themselves consisting of sub-phases, which will guide the developers to plan, manage, control and evaluate courseware development projects. However, a methodology is more than merely a collection of things. It is usually based on some ‘philosophical view’ or a specific theory; otherwise it would have been merely a method.”

2.3 Standards for course development methodology

Standards of CDM discussed in [6] offer solutions to the recurring development problems, and their adoption is beneficial. According to these standards, the courseware should be usable, aesthetically attractive, and educationally effective. More specifically, these attributes should have following characteristics:

- Usability
 - Helps learners to achieve their goals.

- Learners make few mistakes and achieve their goals in a few mouse movements.
- Easy to learn how to use.
- Users like it.
- Aesthetics
 - Easy to assimilate the information on the screen.
 - Visually attractive (multimedia of good quality).
 - Interesting.
- Educational efficiency
 - Allows the learners to navigate the material in a flexible manner.
 - Allows for a variety of types of learning experience, using the full potential of the system.
 - Provides an integrated learning environment .

2.4 Computer-assisted online assessment and feedback

It is widely accepted that when seeking to implement computer assisted assessment (CAA) we should not begin with the technology but with the pedagogy. If CAA is to be used appropriately, then it must be regarded as a range of assessment strategies, including objective tests, formative self-assessment, and so on. In [8] a methodology for online assessment and feedback is designed to be used by module teams to explore opportunities for online feedback and assessment. The methodology is influenced by the initial literature review and by critical consideration and evaluation by practicing academics at the university.

The methodology as shown in Figure 2.1 is made up of three stages:

- Planning and design
- Development
- Performance and evaluation

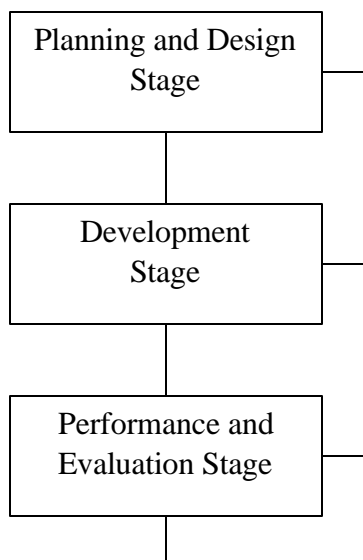


Figure 2.1 *Key stages in methodology for online feedback and assessment*

The planning and design stage

The planning and design stage is concerned primarily with pedagogic considerations. Taking into account the context of the unit/module, appropriate feedback activities and opportunities are designed for learners. The feedback activities are compiled into a feedback plan for the unit/module that specifically tries to satisfy the stated learning outcomes.

The development stage

The development stage is concerned with the creation of resources to support the student experience with online feedback and assessment environments.

The performance and evaluation stage

The performance and evaluation stage is concerned with piloting, performing, and improving the online feedback and assessment activities in an environment where contingency plans are in place to protect the student experience.

2.5 Existing online courseware systems

Our online courseware program, CS.UNR E-Learning, finds its roots in existing popular courseware systems like WebCT, Blackboard, *etc.* The following sections discuss some of these systems

2.5.1 WebCT

WebCT (www.webct.com), officially called “WebCT Campus Edition”, is one of the most popular e-learning systems available to date. WebCT provides a complete, institution-wide e-learning solution. By using WebCT, institutions are able to offer their students the highest quality educational experience and to support their faculty's diverse teaching styles and students' learning needs. In one system, WebCT provides the broadest set of easy-to-use teaching and learning tools for course development, course delivery, and course management. Following are highlights of WebCT:

Course utilities

Utilities include the following: ‘Course Builder’ to help new users to construct essential course content; ‘Designer Map’ (shown in Figure 2.2), which is a control center for the course with links to all course utilities, course appearance settings, tools for managing files for course content, and tools for managing student and their grades; ‘Content Assistant’ where instructors can find resources and content for the course; ‘My

WebCT' (shown in Figure 2.3) module presents a personalized page for the user to add bookmarks and add/remove courses and also includes a language plug-in to show the web-interface in different (supported) languages; and 'Resume Course', which allows students to come back to the page that they last visited.

MYWEBCT RESUME COURSE COURSE MAP RESOURCES LOGOUT HELP						
Designer Map: Designer Options						
Tools To perform one of the tasks below, click it.			Utilities To perform one of the tasks below, click it.			Designer Options Below are the tools, organizer pages, and content modules that you've added to your course. To modify one of these, click it.
Content Assistant	Update Student View	Add Page or Tool	Manage Files	Manage Course	Change Settings	Modify
		<ul style="list-style-type: none"> ● Pages/URLs <ul style="list-style-type: none"> ○ Organizer Page ○ Single Page ○ URL ● Contents & Related Tools <ul style="list-style-type: none"> ○ Content 	<ul style="list-style-type: none"> ● Upload File ● Create ● Edit ● Copy ● Move ● Rename ● Delete 	<ul style="list-style-type: none"> ● Manage Students ● Track Students ● Manage Presentation Groups ● Manage Teaching Assistants 	<ul style="list-style-type: none"> ● Instructor Name ● Language: Design View ● Language: Student View ● Welcome Page ● Course Menu ● Course 	<ul style="list-style-type: none"> ● Homepage ● Calendar ● Chat ● Discussions ● Student Homepages

Figure 2.2 *Designer Map**

Course components

Course components are formed by the course homepage, syllabus, and content modules. The course homepage is customized by the instructor and appears as the first page when the student logs in. The syllabus is used to provide students with important information such as course objectives, textbooks, course requirements, and contact information. The content module presents the course material to the student.

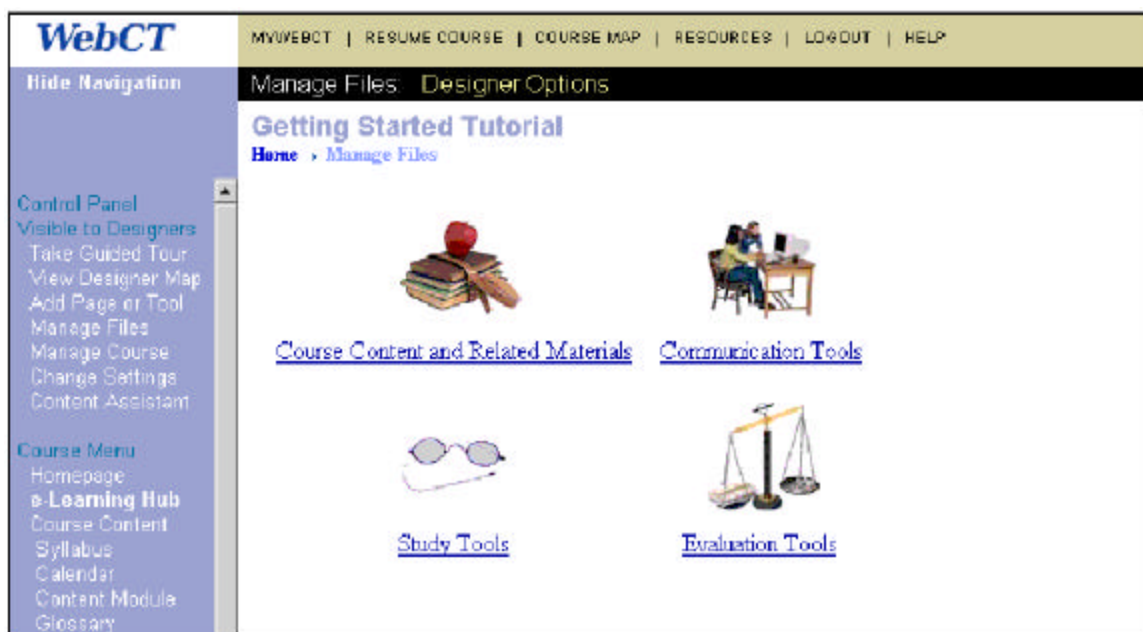


Figure 2.3 My WebCT screen*

There are communication-rich tools, like a white board, and chat and discussion forums. These tools facilitate the creation and automatic making of quizzes, surveys and online tests.

Course tools

Course tools allow distributing and receiving course assignments electronically and can facilitate grading, and commenting on submitted assignments. These tools provide a calendar for managing events and notes using a shared calendar (shown in Figure 2.4). There are communication rich tools like, like a white board, and chat and discussion forums. These tools facilitate the creation and automatic making of quizzes, surveys and online tests.

MYWEBCT | RESUME COURSE | COURSE MAP | RESOURCES | LOGOUT | HELP

Calendar: View Designer Options

Intro. to the Internet
[Home](#) > [Calendar](#)

June 2001
 Date: Month Year
 To compile a list of entries, click [Compile](#).

 To view, add, or edit the daily schedule, click a hyperlinked date below.
Note: All private entries are italicized.

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
View Week						<u>1</u>	<u>2</u>
View Week	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>

Figure 2.4 *Calendar of events**

2.5.2 Blackboard

The Blackboard (www.blackboard.com) teaching and learning environment is widely regarded as industry's course management system on the basis of ease of use, widespread adoption, pedagogical flexibility, and a breadth of intuitive features and functions for either web-enhanced or distance learning. Blackboard's online teaching environment includes four primary areas of functionality: content management, communication tools, assessment tools, and control options.

Content management

The content management tools are the tools for announcements, calendar events, tasks and grades from the multiple courses in one aggregated view known as My Institution Portal as shown in Figure 2.5.. Blackboard allows access to the courses

* Figures are taken from <http://www.webct.com>

through links in the portal. Other tools are used for maintaining personal calendar, address book, user directory, and to-do lists.

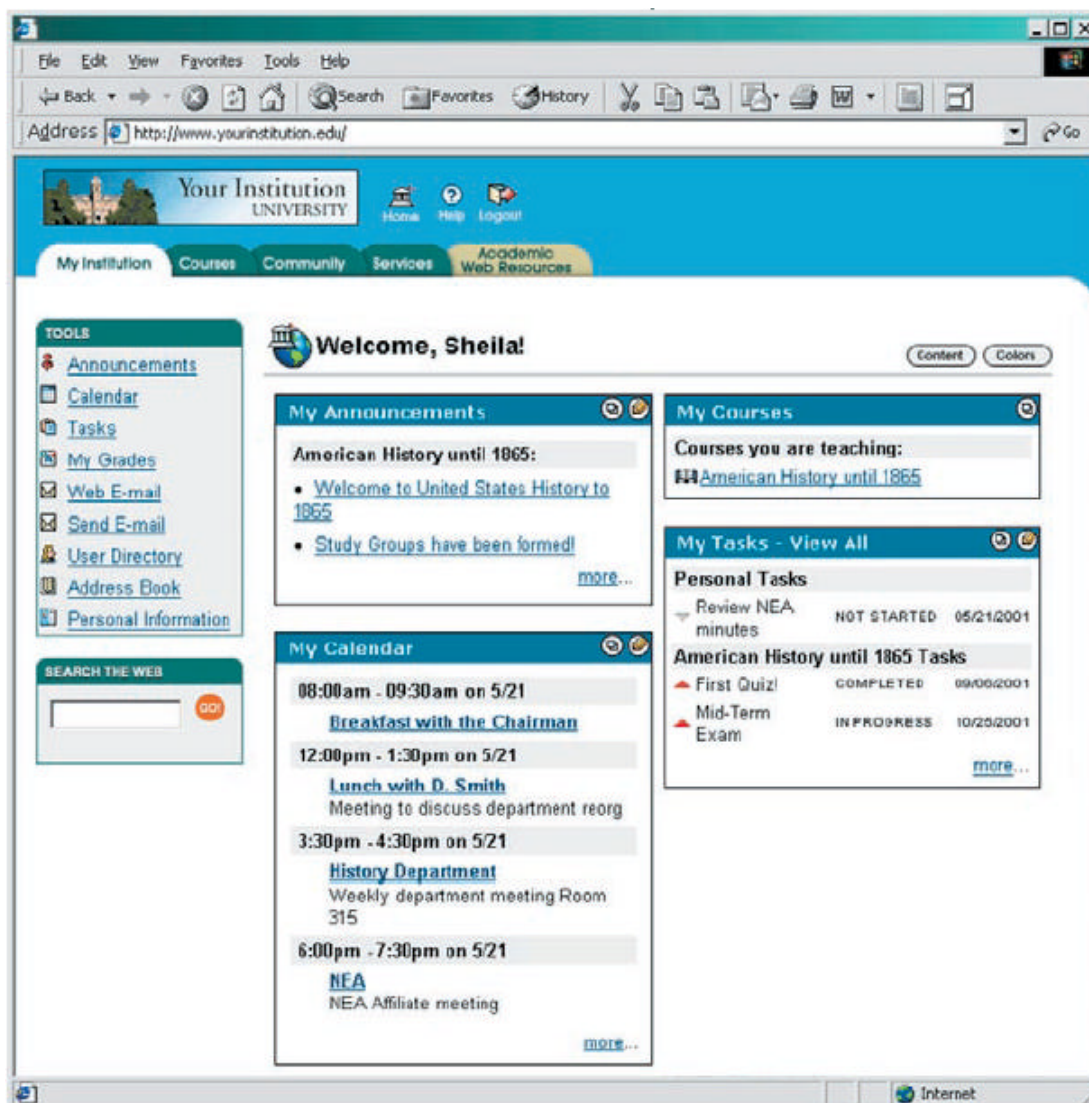


Figure 2.5 “My Institution” portal interface*

Instructor options include tools for maintaining course websites and a folder structure to organize content. Blackboard allows the instructor to author basic content items directly through the form-based user interface or to incorporate existing instructional

* Figures are taken from <http://www.blackboard.com>

content by uploading files. It can schedule and automate the release and recall of content for specific dates and times. It is loaded with tools for creating learning units to define sequential learning paths through content and assessment. The useful Blackboard-enabled course cartridges are available from leading academic publishers *via* a Blackboard cartridge server and allow seamless integration of professional digital course content into any Blackboard teaching and learning environment.

Students can browse discipline-specific information, resources, and communities at a single click. They can customize an academic web resource site to individual needs and preferences. The workspace has direct access to a wide range of journals and training material on how to use the Blackboard products effectively.

Communication tools

Blackboard features a rich set of communication tools for enhanced collaborative learning through discussion boards, real-time virtual classroom interactions, and group communications as shown in Figure 2.6. There are options for creating and managing forums and live on-line text chats, accessing a draw-in shared white board, collaboratively browsing, and graphically marking up web pages in the updated virtual classroom environment. The students can submit their work and assignments through a digital drop box. The instructors can create groups of students for collaborative work, enable protected discussion boards, and file exchanges for each group.

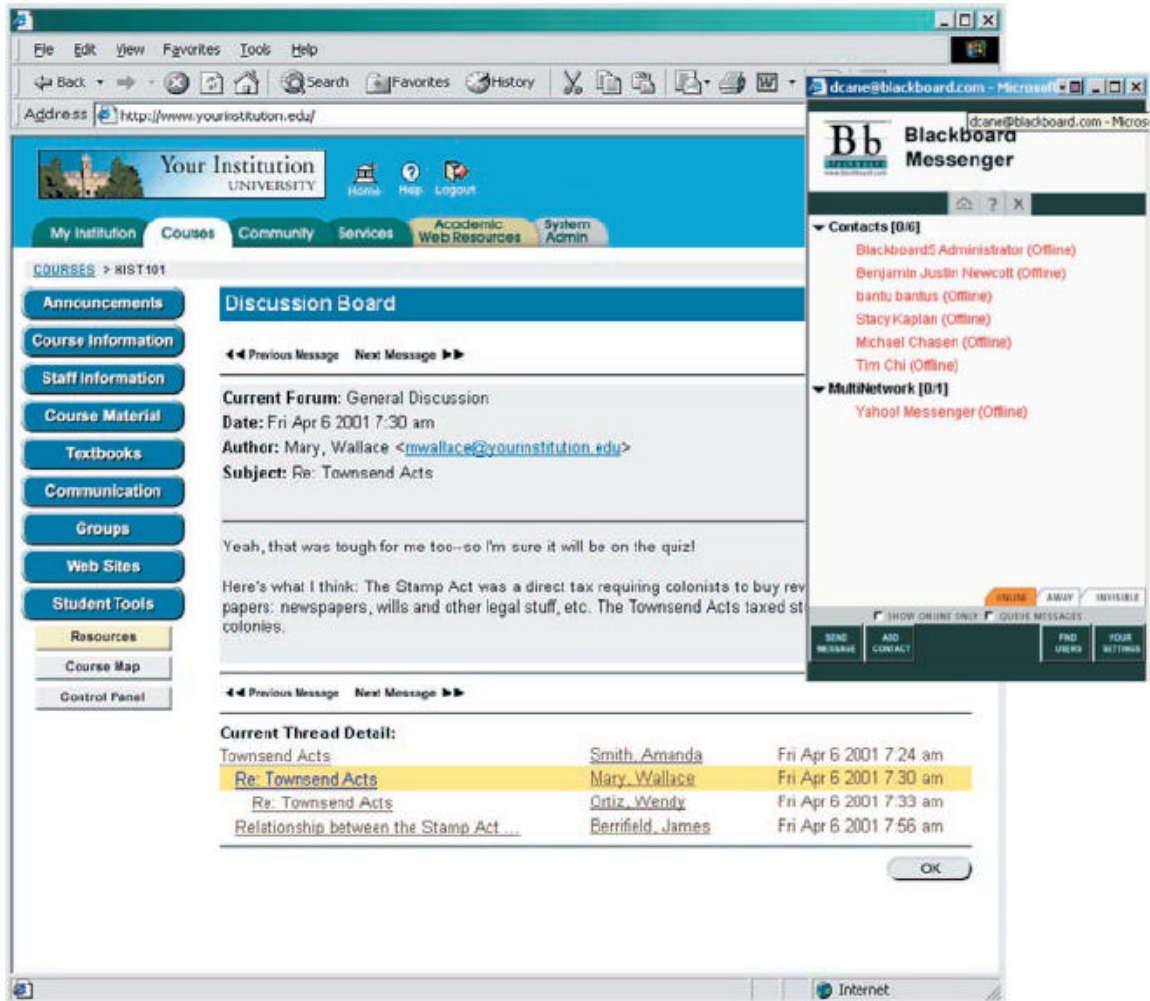


Figure 2.6 *Communication tools**

Assessment tools

The assessment tools allow creating assessment questions with multimedia or other attachments and then creating quizzes and surveys using these questions. These tools facilitate delivery of password-protected tests, timed tests, and instant feedback to the

* Figures are taken from <http://www.blackboard.com>

by date-time.

Control options

The control options allow managing courses through an easy-to-use, robust control panel. These options facilitate easy enrollment or de-enrollment of the individuals or groups of students. Tools are provided to recycle a course between the academic terms by automatically resetting the discussion boards, assessments, and other content areas. Additionally, tracking tools are included for monitoring the student progress, grades, and the course content usage.

2.5.3 Centra Symposium

Centra Symposium (www.centra.com), a virtual classroom, is the most widely used solution for the delivery of live e-learning solutions across the enterprise. Unlike streaming broadcast technology or static Web pages, Symposium enables large groups of dispersed employees, partners, and customers to interact, collaborate, and learn. This product replicates a typical classroom interaction in real-time over intranets, extranets, and the Internet. Following are the features of this program:

Interaction tools

Symposium features IP audio and video conferencing, choice of languages for participants, breakout rooms, and labs for session leaders to break the main session into multiple breakout rooms for the team or individual lab exercises. The leader can "look over the participant's shoulder" and even take control of their applications. It supports multiple presenters and enables evaluation through quizzes and surveys. It has a whiteboard that allows multiple users to interactively markup the whiteboard, including the content created in a breakout-room, which can be saved for later review. The session

leader can control which participants have the ability to speak and interact with the collaboration. The participants can raise their hands, indicate yes/no polling, demonstrate laughter and applause responses, utilize the public/private text chat, and provide anonymous feedback to the leader and presenters. These tools also support peer-to-peer interaction and text chat.

Asynchronous learning

Live and self-paced blended content is presented through the self-paced learning objects that can be included in a live Centra Symposium session. The learning objects can be updated while they are live in session. This kind of integration easily enables blended learning. The events are automatically recorded on the server and can be played back later by registered or guest users. These events can be configured to allow the enrolled participants to access the event content before and after the live session.

Rich content support

Leaders and participants can share any windows application, including their entire desktop or even a remote server, with other participants for IT and software product training. The rich content support allows the session leaders to take the users on a synchronized web tour. The 'Agenda Builder' is a sophisticated agenda creation tool for building web-based evaluation forms and assembling event content into a structured and customized format.

Low bandwidth requirements

Centra Symposium, being thin-client web architecture, has 28.8 kbps dialup support. The events can be configured so that the event content will be automatically pre-cached on the participant's PC. To reduce bandwidth requirements across an organization's wide

area network (WAN), satellite servers can be deployed for strategic and geographically dispersed locations. The participants of the same event will then automatically receive the data and content from their local satellite server. Even if a satellite server goes down during a live session, the users will be re-connected automatically to the next best satellite server available. Centra Symposium events can be run on the top of an IP multicast infrastructure for dramatic reductions in bandwidth utilization across the WAN.

2.5.4 eMind

eMind specializes in e-learning solutions tailored to the securities, insurance, accounting, and banking industries. It has crafted a best-of-breed library of accredited courses, combined with a delivery platform that is solely designed to meet the needs of the financial services marketplace and a suite of Integrated Workflow Solutions for managing compliance requirements.

eMind platform

eMind is constantly updating technology to meet the needs of the customers. The Web-based platform requires no software installations, and updates are automated, eliminating costly technical integration. eMind has developed a proprietary platform that is uniquely suited to financial services clients. The salient features of this platform are corporate messaging, learner transcripts, insurance CE tracker, CFP tracker, performance resource library, downloadable or printable courses, silent logon, calendar, export tool, offline event tracking and reporting, online administrative and reporting tools, company branding, live online instructor-led training, and automated email.

Course engine and authoring tool

eMind's authoring tool is designed to give the power to easily create highly customized courses and curricula from a collection of thousands of learning objects. Using a web browser, it allows for the creation of new learning objects as well as for their arrangement in the course. Access to the tool is restricted to the users that you establish *via* login names and passwords. All progress, launches, assignments, and scores can be reported in Microsoft Excel-based reports that include advanced filtering capabilities. The instructors have the ability to track how their learners are performing.

Knowledge objects

eMind builds its course content in 'Knowledge Objects.' This cutting-edge technology allows for flexibility in a way that course content can be reorganized, topics taken out, and topics added with minimal effort. This modification is available on a per-client basis, so that each client can have a customized version of the same course, without the custom costs. Knowledge objects can be retrieved individually by searching for topics of interest. This provides a reference source and an interactive method of retrieving the course content, and new courses can be created with minimal effort by linking the existing objects.

Offline training

Offline (classroom) training and online courses can be assigned and tracked in one place through the eMind platform. It allows for managing, tracking, and reporting of the offline learning events. eMind administrators can manage all aspects of offline learning, including event scheduling, instructor assignment, and completion status. The learners

are informed about offline events *via* email and/or an online calendar function. The training results for both offline and online evaluations are combined in a single report. These tools also support assignment and tracking of continuing education credits for offline events, where appropriate.

Live instructor

A live instructor can guide the remote classroom participants through the materials of an online course. The streaming video and audio of the instructor appear on-screen beside the course materials. This interactivity allows learners to ask questions and makes polling available.

2.5.5 Macromedia web learning studio

Macromedia Authorware 5.2, Macromedia Flash 5, and Dreamweaver 4 together are not an e-learning system, but are a set of tools to author such systems. These tools address the full range of learning application and delivery requirements, from the text-and-graphics tutorials to the training simulations with audio, animation, and video. More detailed information can be found at www.macromedia.com and here are some of their highlights

Engaging features

- Macromedia Authorware features rich multimedia support, built-in interactivity, and powerful automated navigation tools.
- Macromedia Flash features sophisticated animation, streaming mp3 audio, and low bandwidth vectors.

- Macromedia Dreamweaver features DHTML support, Flash buttons, Macromedia Authorware, Macromedia Flash, and Shockwave integration.

Instructional features

- Instructional templates (shown in Figure 2.7) called as ‘Knowledge Objects’ are pre-built templates with wizards that drastically cut development time. Knowledge Objects accelerate authoring tasks large and small, from creating application frameworks and quizzes to installing fonts or locating a system’s CD-ROM drive.

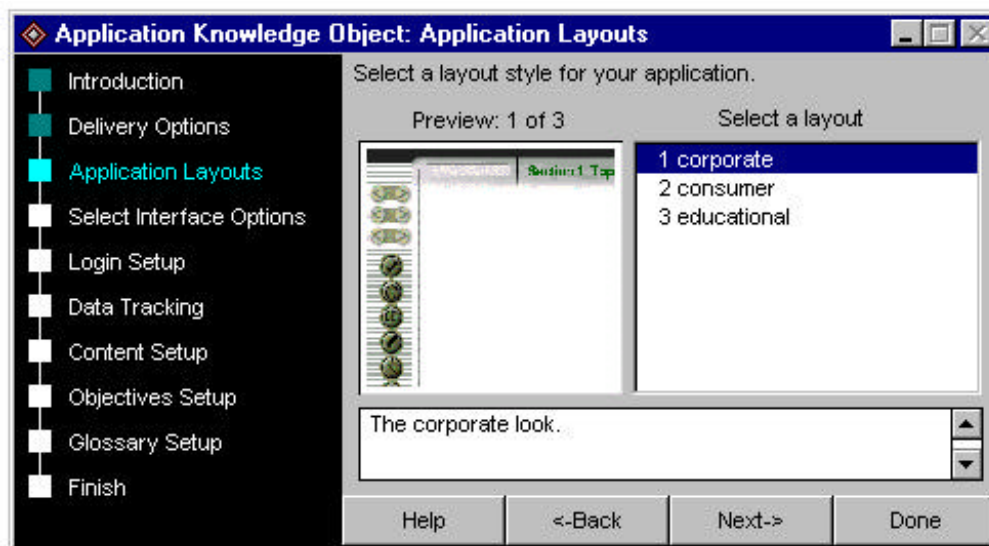


Figure 2.7 *Instructional templates**

- Built-in data tracking is provided through Authorware system variables, which automatically capture and provide easy access to student information. AICC (Aviation Industry CBT Committee) compliant functions and variables allow tracking of student information to open, industry-standard learning management systems.
- Knowledge Tracking provides automatic transmission of student tracking information to open, industry-standard learning management systems (AICC) including Lotus LearningSpace.

- The quiz framework (shown in Figure 2.8) allows the authors to assemble any number of questions into a scored and tracked quiz.

* Figures are taken from <http://www.macromedia.com>

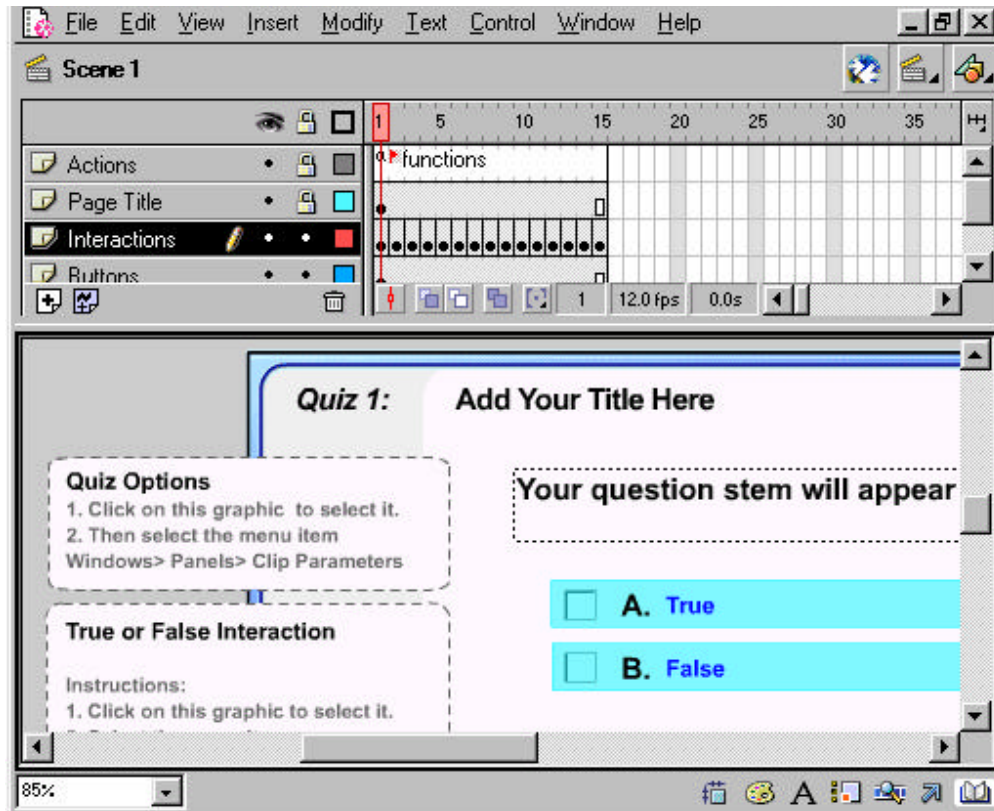


Figure 2.8 *Quiz framework**

2.6 Motivation for CS.UNR E-Learning

In the previous section we saw some of the most popular online courseware systems. None of these systems however provide functionality for computer science courses, such as automated grading and online feedback on programming assignments and their solution files.

Managing highly populated classes is not an easy task, especially computer science classes involving programming assignments. Besides programming assignments there

are quizzes and self-tests, which may form a significant part of the course work. As the population of the class increases, the delay associated with grading increases. There is a need for an automated system that can automatically or with the least amount of human interaction possible grade a significant part of the coursework.

Certainly quizzes with short answers and true or false questions can be graded automatically and some of the programming assignments, which require no human interaction by reading input data from files, can be graded automatically.

Feedback on answers to quizzes or code submitted for programming assignments is another issue that needs to be addressed. This can be a serious issue for larger classes having weekly or biweekly quizzes or assignments where the volume of material to be graded exceeds the time constraints of a grader. This delay often leads to overlapping of assignments that is the failure of the grader to properly grade and return an assignment by the time the student starts working on the next assignment, which may lead to repeated mistakes. Automated system can be programmed to remember the comments on each answer so that the student can get feedback immediately (after or during quizzing).

In addition to the online assessment there are other issues that need to be addressed for simplifying the management of highly populated classes. An instructor should be able to sort emails from different courses automatically and track events through an electronic calendar. A 'class agenda' or to-do list should be provided, through which the instructor can post announcements regarding the upcoming class session, allowing the students to prepare in advance of the class. In addition to these requirements, there is a need to access this program anytime from anywhere.

In summary, we needed a program that would speed up or automate grading of quizzes and programming assignments, provide a user friendly interface, and include simple tools for automatically grading and posting feedback on the solutions submitted by the students. These needs led us to build a web-based program that is fairly portable to all kinds of browsers and operating systems and is an integrated system, whose elements are the tools that accelerate and simplify computer science course management.

2.7 Object-oriented web development

Web developers have long wished for a web development platform that would manage their code, let them separate their business logic from the user interface, reuse their code without having to worry about user interface. Microsoft seems to have provided the solution in the form of the Microsoft.NET platform. ASP.NET, the web department of the .NET platform, is undoubtedly the most advanced web development platform to the date.

2.7.1 ASP (Active Server Pages) and ASP.NET

ASP.NET is not just a new version of ASP. It is a whole new idea, a new way of programming web applications. New features weren't retrofitted into ASP to create ASP.NET. Instead, ASP.NET has been written from the ground up to provide the best possible application framework; however installing ASP.NET will not crash existing ASP applications because the existing ASP pages are still processed by the same mechanism. Following are the key differences between ASP and ASP.NET that led us to the selection of ASP.NET for development:

- ASP scripting is interpreted, but ASP.NET is compiled. Thus ASP.NET is faster than ASP.
- ASP pages are mixed with code and content, which can be a problem for a mixed team with some people designing web pages with HTML and some people doing other coding. ASP.NET has a separate user interface file and what is termed “code behind file”.
- ASP required code for everything no matter how simple (*e.g.*, validating form inputs, maintaining the state of the inputs, *etc*). ASP.NET has simple built-in controls to perform these tasks.
- ASP left browser compatibility to the programmer to handle different browsers using either JavaScript or ASP or both in most cases. ASP.NET can produce HTML content depending on the client browser.
- ASP.NET has built-in support for XML through .NET framework and class libraries, unlike ASP.

2.7.2 Important aspects of the .NET Framework

ASP.NET has a cleaner, event-based programming model, making web development much more like traditional Visual Basic forms programming. This feature results in the average ASP.NET page requiring much less code than an equivalent ASP page, which in turn leads to a great developer productivity and better maintainability. ASP.NET is part of the .NET framework, a new computing platform that simplifies and modernizes application development and deployment on Windows. Following are the most important aspects of the .NET framework:

- .NET is a platform designed for writing Internet-aware and Internet-enabled applications that embrace and adopt open standards like XML, HTTP and SOAP.
- A platform that provides a number of rich and powerful application development technologies like 'Windows Forms', used to build classic GUI applications, and ASP.NET, used to build web applications.
- A platform with an extensive class library, that provides extensive support for data access (relational and XML), directory services, message queuing, and much more.
- A platform, which has a base class library that contains hundreds of classes for performing common tasks like file manipulation, registry access, security, threading, and searching of text using regular expressions.
- A language neutral platform that makes all languages equally powerful. You can use the language you feel the most comfortable and productive with and not face any limitations.
- A platform that doesn't forget its origins and has a great interoperability support for existing components that you or third parties have written, using COM and standard DLLs.
- A platform with an independent code execution and management environment called Common Language Runtime (CLR), which ensures code is safe to run. It provides an abstract layer on top of the operating system allowing elements of the .NET Framework to run on many operating systems and devices.

2.7.3 Common Language Runtime (CLR)

The CLR is the most important part of the .NET architecture and computing paradigm. It is the run-time environment in which applications written in different languages can all exist and function together, providing cross-language interoperability.

The mechanism for cross-language interoperability is provided *via* the concept of Common Language Specification (CLS). A compiler that adheres to the CLS rules can take code written in a specific language and port it to the Microsoft Intermediate Language (MSIL). MSIL cannot be executed directly; the CLR compiles the MSIL using a just-in-time (JIT) compiler into native CPU instructions as it processes the MSIL.

One term that is often mentioned in conjunction with .NET is “managed code”. Managed code is made possible by the CLR. It is just the code that is run under the supervision of the CLR. One of the most important services provided by the CLR as part of the managed code paradigm is the concept of garbage collection. .NET uses automatic garbage collection to reclaim memory. When memory allocated on the heap becomes orphaned, or passes out of scope, it is placed on the list of memory locations to be freed. Periodically, the system runs a garbage collection thread that returns the memory on the heap marked for deletion.

While the runtime is designed for software of the future, it supports legacy technology. The interoperability between managed code and unmanaged code enables the developer to continue to use the necessary COM components and DLLs.

2.7.4 Managed, Unmanaged, and Safe Code – Understanding the difference

Managed code is code that executes under the supervision of the CLR. The user has no control over how the CLR achieves memory management, data type verifications, garbage collection, and many other activities.

Unmanaged code is code that executes outside the context of the CLR. The best example of this is traditional Win32 DLLs like KERNEL32.DLL, USER32.DLL, and the COM components installed on our system. These components decide by themselves how to allocate memory, how memory is released, and how type verification takes place, and many other similar tasks.

Unsafe code is cross between managed and managed codes. It executes under the supervision of the CLR, just like the managed code, but allows you to address memory directly, through the use of pointers, as is done in the unmanaged code. A typical example would be a .NET application that uses the functionality of legacy Win32 DLL, whose exported function requires the use of pointers.

2.7.5 .NET Framework Libraries

The .NET Framework class library is a collection of reusable types that is tightly integrated with CLR. The class library is object-oriented, providing types from which our own managed code can derive functionality.

The .NET Framework types enable you to accomplish a range of common programming tasks. It is possible to create the wrapper classes that provide functionality for XML parsing, database connectivity, and many other common tasks. These classes

can be reused within the same application or by other applications, saving significant time and providing the required reliability.

.NET programs are deployed as assemblies. An assembly is one or more EXE's or DLL's with associated metadata information. The metadata about the entire assembly is stored in the assembly's manifest. The manifest contains information such as a list of assemblies upon which the current assembly is dependant. The concept of assemblies, how they can be deployed privately or publicly, how they can be versioned, *etc.* is complex and can be found in more detail in [3].

2.7.6 Summary – Benefits of ASP.NET

Unlike the unmanaged ASP code, which is always scripted and interpreted, ASP.NET pages are faster and easier to manage as they execute in native machine language and interact with the runtime like any managed application. The .NET Framework provides a plethora of class libraries to cut short most of the tasks. We can use multiple programming languages like VB.NET, C#, Managed C++, *etc.* Web forms and web controls present programmers with an event-driven model similar to desktop application development (like Visual Basic). ASP.NET server side controls intelligently render HTML code appropriate to the target browser. The .NET framework provides advanced crash-recovery. It also provides detection and isolation of rouge code damaging the system, which is termed as unsafe code. The ASP.NET Framework provides a scalable architecture with extensive caching to reduce the amount of identical code run for each website visitor. Debugging is much easier, especially when the programming is done using an IDE like Visual Studio.NET, which also provides good support for database and

XML. MSIL and the new “assembly” model make deploying applications and versioning complexities an issue of the past.

Figure 2.9 shows the typical sequence of calls made during a request-response cycle for an ASP.NET web page. Notice that the first request-response cycle is shown by steps 1 through 6, and subsequent calls are displayed as steps 7-10. Since ASP.NET uses compiled code instead of interpreted code, subsequent requests will check to see if there has been any change in the code. If not, it will utilize the script engine cache to access the compiled version of the code.

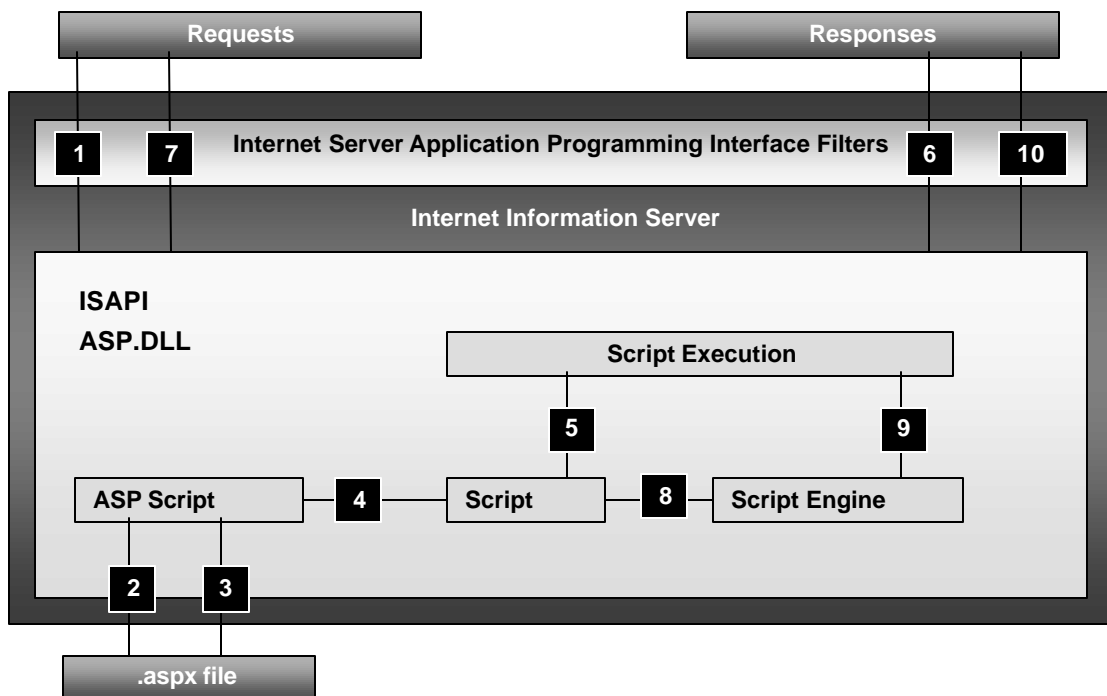


Figure 2.9 Request-response block diagram

Chapter 3

CS.UNR E-Learning Features

From the different user's point of view the Online Courseware Manger is divided into 4 different workspaces. These are the instructor's workspace, the student's workspace, the teaching assistant's workspace, and the administrator's workspace. Currently only the instructor's workspace has been completed. The following sections highlight the features of these workspaces. The TA and administrator workspaces are discussed in Chapter 4, Future Work.

3.1 Instructor's workspace

An instructor can control all of the courses from the instructor's workspace (shown in Figure 3.1). A dropdown list of courses the instructor is teaching is presented at the top right corner of the window. The instructor must select the course and activate it in order to work with that course. This course remains the active course throughout the session.

3.1.1 Course tools

A. Add course

When the instructor signs in for the first time, the list of courses is empty, and the instructor is required to add a course to the system. This can be done through the Add Course option (shown in Figure 3.2). The required fields are course name and number,

department offering the course, and the course start, end dates. Clicking the calendar button (📅), opens up a calendar to be used in selecting dates. Dates cannot be entered manually in the read-only text box.

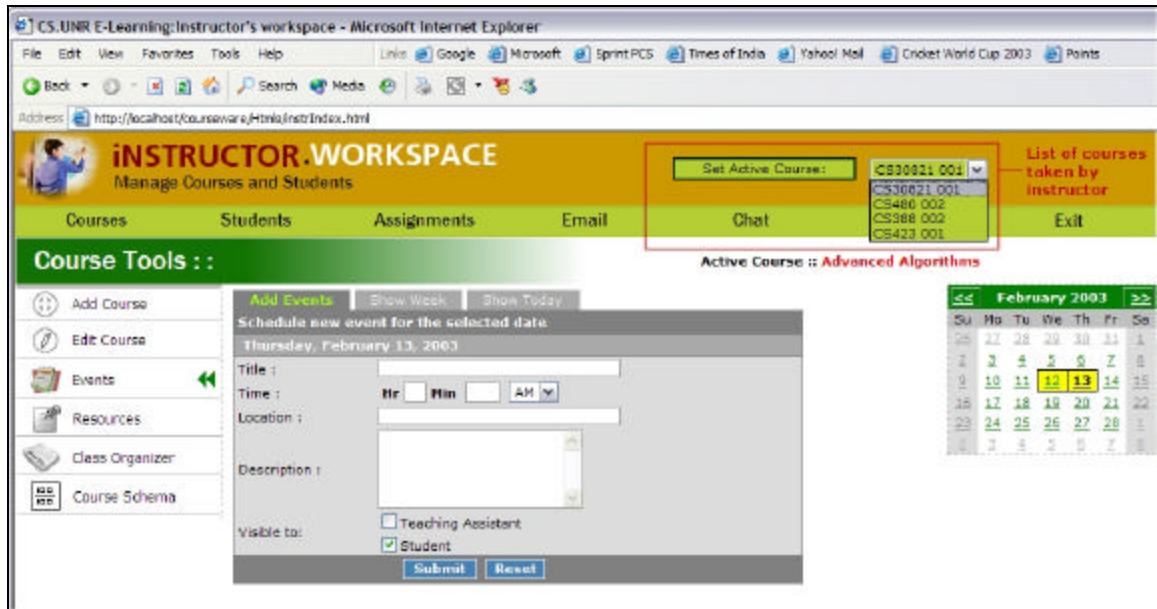


Figure 3.1 Instructor's workspace and list of courses registered by instructor

The syllabus file, which can be in any format (PDF, DOC, Text or HTML), is stored in binary format in the database. When the user submits the form, a regular expression check is performed on all the inputs to validate them, and if found incorrect, the user is notified so with an appropriate error message at the bottom of the web form.

Addition of a course consists of a transaction, which includes the following tasks:

- Add a record to the list of courses in the system.
- Create the course table with its name derived from the course number and section number, if provided.
- Create a class list/schedule based on the selected start and end date. (We discuss more on the Class Schedule in section 3.1.1.)

>> Course tools :: Add course

Course Number :

Course Name :

Section of the course :

Department :

Call Number :

Course Description:

Course website :

Course Email :

Course Timing :

Meeting place:

Course Start date:

Course End date:

Credits :

Upload Syllabus File :

Figure 3.2 Add course form

If any of the above tasks fails to complete successfully, the entire transaction is aborted and the course is not registered into the system. After successful registration, the course is available to the instructor when they next log in.

B. Edit course

The course information for an active course can be edited using the Edit Course option (shown in Figure 3.3). If a new syllabus file is uploaded, then it will overwrite the existing syllabus file. The option for deleting the course is also provided on the same form. Deleting the course will perform a major deletion cascade throughout the database and all related records from all the tables will be deleted.

>> Course tools :: Edit course

Course Number :

Course Name :

Section of the course :

Department : ▼

Call Number :

Course Description :

Course website :

Course Email :

Course Timing :

Meeting place :

Course Start date :

Course End date :

Credits :

Upload Syllabus File :

<< February 2003 >>						
Mo	Tu	We	Th	Fr	Sa	Su
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	1	2
3	4	5	6	7	8	9

Figure 3.3 Edit course form with calendar shown for updating dates

C. Events Manager

The Events Manager tool is one of the most critical tools required to keep track of important deadlines and upcoming events. Each day of the calendar can have any number of events, and all these events can be added by using the Add Events option (shown in Figure 3.4). Event information consists of event date, time, title, location, and description. An option is provided to hide these events from the students and/or TAs so that the instructor can use this tool to set instructions for TAs that are not visible to the students or as a personal organizer.

Other options are Show Events (shown in Figure 3.5), which shows the events for a week containing a date selected from the calendar on the right hand side of the events

table, and Show Today's Events (shown in Figure 3.6), which shows the current day's events if any are available.

The calendar is coded with a day in pink indicating today, a day in green indicating a selected date on the calendar, and a day in yellow indicating a day containing scheduled events.

The screenshot shows two parts: a form on the left and a calendar on the right.

Form: Add Events

- Buttons: Add Events, Show Week, Show Today
- Title: Schedule new event for the selected date
- Event date: Wednesday, February 19, 2003
- Title: [Text Input]
- Time: Hr [Input] Min [Input] AM [Dropdown]
- Location: [Text Input]
- Description: [Text Area]
- Visible to:
 - Teaching Assistant
 - Student
- Buttons: Submit, Reset

Calendar: February 2003

Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	1
2	3	4	5	6	7	8

Figure 3.4 Add Events

The screenshot shows the 'Show Week' view of the calendar interface.

Show Week

- Buttons: Add Events, Show Week, Show Today
- Events scheduled for the week of selected date
- Wednesday February 12, 2003
 - 3:30 PM Project Meeting [Delete](#)
- Thursday February 13, 2003
 - 4:00 PM : Midterm 2 next week [Edit](#) [Cancel](#)
 - Exam on regular classroom
 - Access Level:** Student only

Figure 3.5 Show events for the week of selected date

The screenshot shows the 'Show Today' view of the calendar interface.

Show Today

- Buttons: Add Events, Show Week, Show Today
- Events scheduled for today
- Thursday February 13, 2003
 - 4:00 PM : Midterm 2 next week [Edit](#) [Cancel](#)
 - Exam on regular classroom
 - Access Level:** Student only

Figure 3.6 Today's events

The user control designed for showing events has two sets of properties: the start date and end date of the period for which the events must be queried from the database. This duration can be set internally using Show Type, which can take any of the following values: today, any particular day selected on the calendar. These properties provide a flexible way to program this control for multiple purposes.

The initialization process of this control requires that the start and end date properties are set properly, otherwise an exception is thrown. The default value for the Show Type property is the current day.

D. Resource Manager

Course materials such as presentations, notes, and hyperlinks for reference materials can be posted using the New Resource option (shown in Figure 3.7). A resource can be either a file or text manually entered by the instructor. Before creating a resource, the user needs to decide the category of resource from a dropdown list of resources, which contains the following choices: presentation, hyperlink, descriptive, or other files. The presentation and other files require file upload to the database. Hyperlink and descriptive types require manual text entry. Maximum file size that can be uploaded is decided in the setting files for the application (default is 2MB). All the fields on the New Resource form are required. Upon selection of the category, the control loads appropriate input fields and the mechanism to validate the inputs.

The Show resources option (shown in Figure 3.8) allows the instructor to see or delete the resources for all of the courses taught by him/her. The resources can be searched for a specific keyword in the title or can be filtered for a specific course using

the list of courses. Within the list of resources, sorting is provided by title, date posted, type, and course in ascending or descending order, making it easy to find a specific resource. Clicking on the resource title opens up that resource. If the number of resources goes beyond 20, they are shown on separate pages with links to other pages appearing at the bottom of the list.

Figure 3.7 *New resource*

<u>Title</u>	<u>Course</u>	<u>Type</u>	<u>Date Posted</u>	<u>Delete</u>
my second presentation	CS30821	Presentation	10/12/2002	Delete
my first information	CS30821	Descriptive	10/12/2002	Delete
my second information	CS30821	Descriptive	10/12/2002	Delete

Figure 3.8 *Show resources*

E. Class organizer

The Class Organizer is an agenda or to-do list for a course, organized in small units of class sessions. The instructors can use this organizer to post announcements related to the topics to be covered in the class in advance, allowing the instructor to organize the class better and also helping the students to prepare before each class day. Creation of the Class Organizer is a part of the process of creating the corresponding course. Classes

are created for each day in a week that a course is offered between the start and end date. The classes can be added or deleted using the option Add Class or Delete. The tasks, which are added to the class using the Add task option, appear immediately after successful addition in a list of tasks (shown in Figure 39) scheduled for that class. Resources used as reference material for a task are listed by checking the checkbox labeled Attach Resource. Whenever a class is deleted from the list or a new class is added to the list, the class numbers are rearranged to form a continuous sequence (shown in Figure 3.10).

Title	Resource	Edit	Delete
Read chapter 3 before class	View	Edit	Delete
Assignment 4 discussion is class	View	Edit	Delete
sorting algorithms and complexity	View	Edit	Delete

Figure 3.9 *Class Organizer*

The tasks in the Class Organizer are different from the events in the events manager module. These tasks cannot be hidden from the students or TAs and are limited to the scope of the class session, whereas the events are valid over the length of the entire course. This is one of the modules that distinguish this application from WebCT or similar programs.

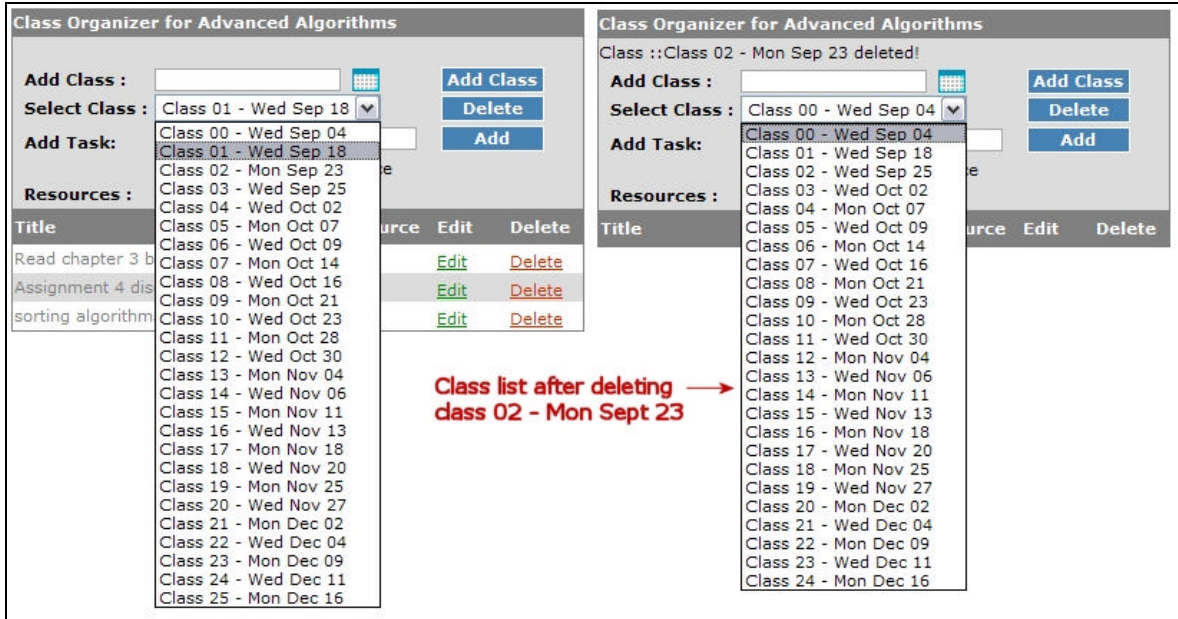


Figure 3.10 Rearrangement of classes after deletion

F. Course Schema/composition

The Course Schema is a visualization (shown in Figure 3.11) of the tentative composition of the grading in a course. The categories are predefined and cannot be altered by the instructor.

Course Tools :: Course Schema	
Examinations :	10 %
Written Assignments :	20 %
Programming Assignments :	30 %
Quizzes :	10 %
Attendance :	10 %
Projects :	10 %
Presentations :	10 %
Oral Examinations :	0 %
Other :	0 %
<input type="button" value="Submit"/> <input type="button" value="Reset"/>	

Figure 3.11

Course Schema or course composition

3.1.2 Student Management Tools

A. Show students

Show Students is the tool used to list all the students in the course. The list shows first name, last name, and email address of each student (shown in Figure 3.12). It can be sorted by first name, last name, or email address in descending or ascending order. A link is also provided to delete a student from the course. The students appear in this list after they register for the course when it is open for registration; thus there is no need for the instructor to add the students manually to the course. If the number of students in the course exceeds 20, then the list is divided into different pages, and links to other pages appear at the bottom of the list.

List of students for Advanced Algorithms ::

First Name	Last Name	Email	Delete
John	Smith	john@cs.unr.edu	Delete
Kishor	Waikul	waikul@unr.nevada.edu	Delete
Sachin	Tendulkar	sachin@yahoo.com	Delete

[<>](#)

Figure 3.12 Student list from Show Students option

B. Turn on/off course registration

The course registration on/off switch is designed to allow the students to register themselves into the course (shown in Figure 3.13). The course self-registration can be turned on or off by the instructor. In the student's workspace there is an option called 'Search Open Classes'. This option allows students to see the list of courses that have their registration turned on. This feature eliminates the need of adding students by the instructors. When a course is created, the registration is turned off by default.

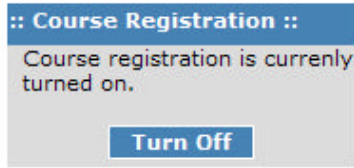


Figure 3.13
Course registration switch

C. Manage Grade Book

The Grade Book module is formed by two sub modules: Edit Grades and Grade Formula. The Grade Formula module (shown in Figure 3.14) shows a list of evaluation items (or course components) that are used to calculate the final grade (e.g., written assignments, programming assignments, quizzes, *etc.*) Each of the course components can be assigned a specific percentage of the final grade using this form. The sum of percentiles is validated so that it doesn't exceed 100%. These percentiles will be used to calculate the final grade of the student.

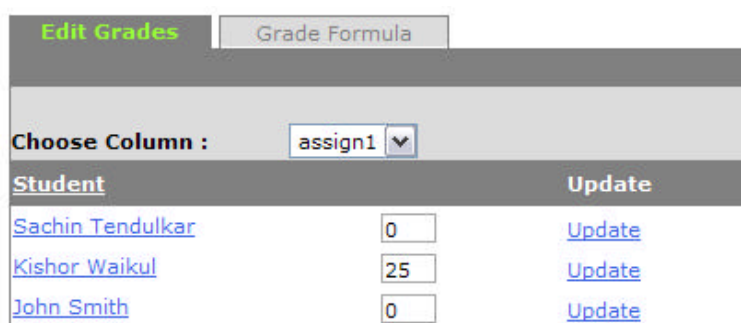
Course component	Percent	Update
assign1	<input type="text" value="3"/>	Update
assign2	<input type="text" value="5"/>	Update
assign3	<input type="text" value="4"/>	Update
assign5	<input type="text" value="4"/>	Update

Figure 3.14 *Grade formula*

Edit Grades is used to update the grades of each student manually (shown in Figure 3.15). Although the grades for the quizzes and programming assignment do not require this module because those are updated either automatically (for quizzes) or through another module (for programming assignments), course components like class presentations, written assignments, and examinations require the Edit Grades module.

The points entered here are cross checked against the maximum points allowed for the corresponding component.

A list of the students in the course appears in this module, and a particular course component can be selected from the drop down menu at the top of list for updating. If there are more than 20 students in the list, it will be divided into pages with 20 students per page and links to other pages appearing at the bottom of each screen. A sorting feature is provided to sort the selected course component according to grade or name in ascending or descending order.



Edit Grades		Grade Formula
Choose Column : assign1		
Student		Update
Sachin Tendulkar	<input type="text" value="0"/>	Update
Kishor Waikul	<input type="text" value="25"/>	Update
John Smith	<input type="text" value="0"/>	Update

Figure 3.15 *Edit Grades*

3.1.3 Assignment Tools

A. Add questions

The Add Questions module allows the instructor to create a quiz, test, or assignment by adding one of the three types of questions: multiple choice, true/false or fill in blank. The questions are stored in a question bank, which may later be accessed to build course components as needed. Each type of the question has an associated feedback input, which appears after evaluation. The feedback specifies the correct answer and may also contain explanatory information.

All the questions have a difficulty level associated with them, which may be used to help an instructor has to create a well-balanced quiz. The questions can be categorized as an assignment, a quiz, or a self test. These categories can be used to automatically filter questions while creating quizzes and examinations.

Each multiple-choice question allows four answers, only one of which is correct. A layout setting allows the answers to be arranged either vertically or horizontally (shown in Figure 3.16 a). The correct answer is indicated by choosing one of the four radio button (A, B, C, D). Fill-in-the-blank questions have text input, where the correct answer for the question is to be entered (shown in Figure 3.16 b). True/False questions are provided with two radio buttons (true and false) to set the correct answer for the question (shown in Figure 3.16 c).

B. Edit questions

The Edit Questions module allows for editing question content. A list of the questions is displayed with a link to edit each question. A search option is provided to search for particular text inside the text of the question. Clicking the Edit link opens up the same form that was used to add the question with all inputs as previously set. Clicking the Update button after making changes will update the selected question in the question bank and return the user to the list of questions from which another question may be selected.

Adding question for course : Advanced Algorithms

Question :

Difficulty level: Easy Intermediate Difficult

Category:

Points:

Negative Score: Allow Do not allow

Choice A:

Choice B:

Choice C:

Choice D:

Answer : A B C D

Answer Layout: Horizontal Vertical

Feedback :

Figure 3.16 (a)

Adding question for course : Advanced Algorithms

Question :

Difficulty level: Easy Intermediate Difficult

Category:

Points:

Negative Score: Allow Do not allow

Answer Text:

Feedback :

Figure 3.16 (b)

Adding question for course : Advanced Algorithms

Question :

Difficulty level: Easy Intermediate Difficult

Category:

Points:

Negative Score: Allow Do not allow

Answer: False True

Feedback :

Figure 3.16 (c)

Figure 3.16 Add Question (a) Multiple choice (b) Short answer (c) True/false





C. Manage Files


Programming assignments require the students to upload solution files. For the ease of manipulation, these files are stored in the database instead of being stored as physical files on the server. An automated grading module (to be developed) would download these files from the database server for compiling and running, and then the log/result files will be saved in the database for verification of results. The instructor or TA can

post feedback on the submitted code, and the students can see these comments online as well see Section 3.1.3 for more description.

The Manage Files module shows a list of students in the course for a selected programming assignment. The assignment is selected from a drop down menu of the course components (shown in Figure 3.17). For each student, the score for that assignment is shown along with an icon indicating the status of the submitted assignment.

The status icons are as follows:

-  On-time submission graded (Green)
-  Late submission graded (Green)
-  On-time submission not graded (Red)
-  Late submission not graded (Red)

The Show Files button  loads the list of files for that particular assignment on the right side of the list. Selecting a particular file will load details about file such as timestamp of submission, size of the file, a link for previewing the file as the student will see it after posting feedback (Section 3.1.3.E), and another link to post the feedback on the file (Section 3.1.3.D). The timestamp of the latest submitted file determines whether the assignment was a late submission or not. Whenever a comment is added to a single line of the file, the assignment is marked as graded. The points can be updated by using the Update button. While updating the points, they are verified against the maximum possible points if that value is available. As shown in the Figure 3.17, student names appear as links associated with their email addresses. Just below the Search User button, information about the assignment (due date and maximum points) is shown.

The screenshot shows a web application interface for managing files. At the top right, it says "File Owner:: waikul". Below this, there is a section titled "Files for assign2" with two entries: "reportDlg.cpp" with 0 comments and "pc_info.txt" with 6 comments. The main area features a search bar labeled "Search User" and a dropdown menu set to "assign1". Below the search bar, it displays "Due: 1/23/2003" and "Max Points: N/A". A table lists students with their points and update options:

Student	Points	Update
Kishor Waikul	25	Update
Sachin Tendulkar	0	Update
John Smith	0	Update

At the bottom left, there is a small "1" indicating a page number.

Figure 3.17 *Manage Files*

D. Feedback on files

Feedback on files is one of the distinct features of the program. This module allows the teaching assistant or instructor to select a line of the uploaded file on which to post a comment (shown in Figure 3.18). All distinct comments already posted for that assignment are listed in a pull-down menu. To enter a comment that already exists in the database, the user selects that comment from the list, and it is copied to the text box provided for typing the comment. Text pasted in this manner can be easily modified. The lines containing comments are shown in red to distinguish them from the other lines in the file.

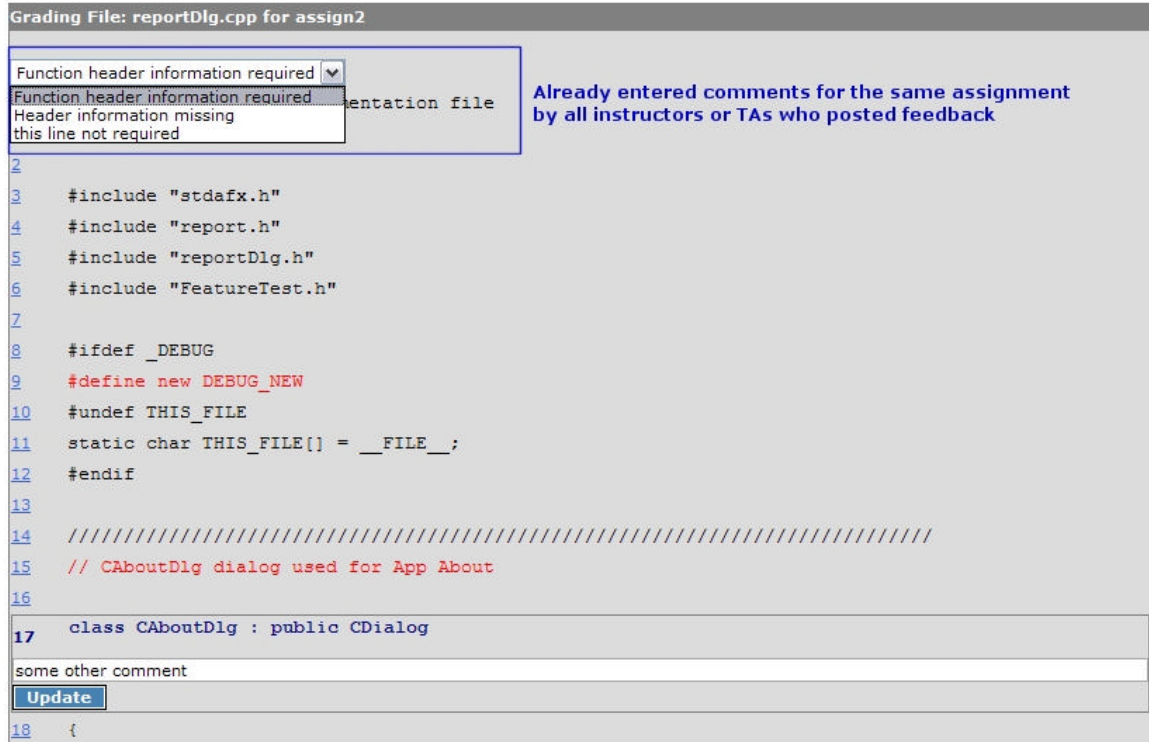


Figure 3.18 *Posting feedback on the solution files*

E. Preview files

The Preview Files module allows an instructor or TA to preview the commented files as they will appear to the students (shown in Figure 3.19). The lines containing comments are posted are shown with a gray background box. The original code is shown in bold blue text, and the comments are shown in green text. Using the Download button, the original code can be downloaded.

This module also shows additional information about the file such as the assignment for which the file was uploaded, the name of the TA or instructor who last graded the file, date graded, *etc.* This information is useful when the same file is being graded by more than one TA or instructor.

File name: reportDlg.cpp	Last Graded by : waikul on Feb 14 2003, 11:25:29 AM	Download Original
--------------------------	---	-----------------------------------

```

0 // reportDlg.cpp : implementation file
1 //
2 // Header documentation missing
3
4 #include "stdafx.h"
5 #include "report.h"
6 #include "reportDlg.h"
7 #include "FeatureTest.h"
8
9 #ifdef _DEBUG
10 #define new DEBUG_NEW
11 // Document the use of this constant
12 #undef THIS_FILE
13 static char THIS_FILE[] = __FILE__;
14 #endif
15
16 // CAboutDlg dialog used for App About
17 // Function header documentation required
18
19 class CAboutDlg : public CDialog
20 {
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Figure 3.19 *Preview commented file*

3.2 Student workspace

The Student Workspace is currently under construction and has only two modules fully implemented. Features of these modules are discussed in the following sections, and the modules that will be added soon are discussed in the next chapter.

3.2.1 Course Tools

A. Registration

The course registration module is the first module that each student will have to use in order to register for the course. This module shows a list of the courses open for registration (shown in Figure 3.20). The courses can be set open for registration by the instructor using the option discussed in section 3.1.2. The courses for which the student

is already registered are not shown in the list. The More Info link takes the student to the detailed course information page (shown in Figure 3.21), which includes a description of the course, name and email address of the instructor, course website address, schedule, and location. Clicking the Register link registers the student into the course. When the student next logs in, this course will be shown in a dropdown list of the courses taken by the student (similar to the instructor's course list in Figure 3.1).

List of courses currently open for registration:		
Course Name		
CS30821 Advanced Algorithms	More Info	Register
CS423 Compilers construction	More Info	Register

Figure 3.20 List of the courses open for registration

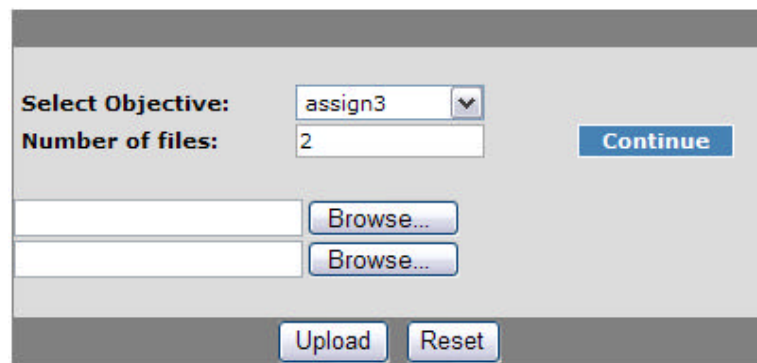
Course :: CS30821 Advanced Algorithms	
Description :	<input type="text"/>
Section :	001
Call Number :	
Department :	Computer Science
Credits :	3
Start Date :	Friday, August 16, 2002
End Date :	Monday, December 16, 2002
Meeting Place :	SEM 233
Time :	
Website :	http://www.cs.unr.edu/cs30821
Email :	cs30821@cs.unr.edu
Instructor:	Kishor Waikul
Syllabus:	Click here

Figure 3.21 Course information

3.2.2 Assignment tools

A. File Upload

The programming assignments require the students to upload the files to the system for evaluation. In this module (shown in Figure 3.22) the student selects the assignment from the drop down menu and enters the number of files to be uploaded. Once this number is entered, that number of file input boxes are presented to the user. When the files are uploaded, a message is shown to the student to reflect the success of the uploading procedure.



The screenshot shows a web interface for file upload. It features a 'Select Objective:' dropdown menu with 'assign3' selected. Below it is a 'Number of files:' input field containing the number '2'. To the right of these fields is a blue 'Continue' button. Underneath, there are two empty text input boxes, each with a 'Browse...' button to its right. At the bottom of the interface, there are two buttons: 'Upload' and 'Reset'.

Figure 3.22 *File Upload for assignments*

Chapter 4

Conclusions and Future Work

4.1 Conclusions

Currently, the instructor's workspace is implemented as described in the previous chapter. This application is an effort toward providing an integrated environment for course management. This program has the potential to speed up the overall grading process, through a fast feedback posting mechanism. This speed up will benefit highly populated computer science courses, especially those that require weekly quizzes and programming assignments. Although much has been completed, the work is far from over because the student, TA, and the administrator workspaces have yet to be implemented. The following sections briefly discuss the projected features of these workspaces.

4.2 Future work

4.2.1 Instructor workspace

In addition to the modules described in Chapter 3, the following are some of the modules still under construction or to be developed. Subsequent sections discuss the essential functions of those modules.

A. Create assignments

The Create Assignments module will create assignments using already existing questions in the database. These assignments can be labeled as quizzes, tutorials, or self tests. Each group of questions will have some properties associated with it, such as duration, date/time of publishing, expiration date/time, IP addresses from which students can complete the assignment, format or layout of quiz, *etc.*

B. Grading daemon

Grading Daemon would be an operating system service acting as an interface between the compiler for programming assignments and our web-based program. The compiler, which is a Linux machine, is separate from the web server. The daemon will download the solution files from the database and compile/run the code. The results files and log of events will be uploaded back to the database for further evaluation by the instructor/ TA.

C. Email options

A third party module to send/receive emails for the course can be used instead of a conventional email program for better, integrated manageability. This module could be automatically loaded with student email address and email-lists.

D. Discussion Board

A Discussion Board will feature options for the instructor to create and initiate discussion boards and to post or delete messages.

E. Course website manager

The Course Website Manager module will provide a content manager for the course website. The instructor will be able to place content on the website, such as upcoming

events, resources, and useful information related to course. In short, this manager will be a shortcut to all the information related to the course that is presented by the instructor.

F. Help

The Help module will contain extensive help about the user interface of the instructor's workspace and its contained modules.

4.2.2 Student workspace

We have already seen two of the options from the student workspace (course registration and file upload). Additional tools that will be incorporated soon are as follows:

A. Grade book

Using the Grade Book, students will see their grades for all the course components for the active course and possibly their current grade ranking in the class.

B. Assignments/quizzes

The Assignment/quizzes module will show a list of previously graded assignments, quizzes, or tutorials. This useful tool would help students prepare for future quizzes or tests by reminding them of what mistakes were made in the previous assignments.

C. Manage files

The Manage Files module will show students the list of files owned by them. These files are grouped by assignment and can be deleted or downloaded. The students can also see online the comments posted on their submission.

D. Test run assignments

The Test Run module will allow the students to test their programming assignments before making final submission for evaluation in exactly the same way the graders will run the program. Using this option, students can run their programs with the sample data provided by the instructor or download/view the sample data for local use.

E. Answer quiz

The Answer Quiz module will deliver the current quiz to the student. The student can answer the quiz over the duration of the quiz specified by instructor, and in the most cases the results along with feedback on the answers will be shown immediately after the quiz is submitted.

F. Discussion Board

Using the Discussion Board, students can participate in a discussion forum initiated by the instructor or TA. The students can post new messages or browse through the replies posted by other students.

G. Help

The Help module will provide useful help on the user interface of the student workspace and the modules it contains. This module will also allow the students to provide feedback on the system which can be used to improve the quality of the program.

4.2.3 Teaching Assistant workspace

The Teaching Assistant Workspace is a subset of tools or modules from the instructor workspace and contains only the tools that are required for evaluation of submitted assignments and answered quizzes. TA modules are: grading daemon, feedback on code

files, grade book, and discussion board. Similar to other workspaces, help topics are provided on the user interface.

4.2.4 Administrator workspace

An administrative workspace is designed to provide total control on the entire system. The administrator can issue temporary passwords, perform online database management, *etc.* More administrative controls will be added as the system develops.

Bibliography

- [1] David Yaskin, Stephen Gilfus. Blackboard 5, Introducing the Blackboard 5: Learning System. www.blackboard.com. 2002
- [2] Consuming ArcIMS Image Services using .NET Client Web Application, An ESRI White Paper .October 2002
- [3] Richard Anderson, Brian Francis, Alex Homer, Rob Howard, Dave Sussman, Karli Watson. *Professional ASP.NET*. Wrox Press, Birmingham, UK, 2001
- [4] Z. Zhiting, Cross-Cultural Portability of Educational Software: A Communication-Oriented Approach, PhD thesis, University of Twente, 1996
- [5] J. Bourdeau, A. Bates, Instructional Design for Distance Learning, *Tutorial Notes within the Fourth International Conference in Intelligent Tutoring Systems (ITS'96)*, Montreal, Canada, June 1996.
- [6] S. Retalis. A courseware development methodology for Open and Distance Learning, In *Proc. CAISE'97* Barcelona, Spain, May 1997.
- [7] D. Avison, G. Fitzgerald. *Information Systems Development: Methodologies, Techniques and Tools*. McGraw-Hill, Berkshire, England 1995.
- [8] Developing a methodology for online feedback and assessment Julie Hanson, Clare Millington and Madeleine Freewood, In *Online Proc. 5th International Computer Assisted Assessment*, Leicestershire, UK, July 2001.
- [9] A.D. Marsha and S. Hurley. The Design, Development and Evaluation of Hypermedia courseware for the World Wide Web. *Multimedia Tools and Applications : An International Journal* (3,1): 5-31, 1996