eClass: Synthesis of an Electronic Course Framework

An Interactive Qualifying Project

Worcester Polytechnic Inststitute

Authored by:

Brian Sperlongano
Ali Taheri

Advisor:

Professor Raymond Hagglund

Project Approved: May, 2000
Project Completed: March, 2001

Abstract

As an emerging educational trend, distance learning presents various challenges and opportunities in an increasingly technological society. This project looks at the issues involved with the creation of an electronic distance learning medium, or "eClass." In completing this project, an eClass implementation is created for ES1020: Introduction to Engineering.

Table of Contents

I. Introduction

Distance learning is an emerging trend in education and with it comes the need for new and more effective forms of delivering the traditional classroom material. With the equally parallel emergence of the Internet and electronic media, the clear solution to this problem is to create courseware in a completely electronic form. This type of learning opens up the educational possibilities of a "global classroom," where students from even remote corners of the world can sit together and share knowledge in the same virtual classroom. The purpose of this project is to create such an entity, which we have given the name "eClass." There are a number of challenges that go along with such a proposal. Very important is the professor-student interaction; the professor who is traditionally central to a course, does not directly interact with the students.

This project encompassed three academic terms. The first term was used for observation, as we observed the various problems inherent in a traditional classroom. Additionally, the first term was used to orientate ourselves with the class we were working with, *ES1020: Introduction to Engineering*. The second term was used to analyze different software design options and implement an appropriate solution along with gathering all the material for the eClass. This included videotaping lectures, and creating all of the coursework in electronic format. The third term was where it all came together, and example problems and lessons were added in to make eClass a completely usable course on a single disc.

II. Term 1: Observation

The intention of the first term of this project was to audit the class to come up with the best way to implement a blood-and-flesh class into electronic form. During this term, we did some initial sample work and began to explore all of our various options for electronic implementation. By far, however, the most important thing that was done was to gain an understanding of the class and how students interacted with it.

Class Overview

ES1020 is a class intended to orient first-year students to the tools and methods of engineering. The course description, as written in the WPI 2000-01 Undergraduate Catalog notes: "Students gain actual engineering experience by working on an engineering problem which has been selected from a professional work experience. Student teams are formed and are assigned the entire problem or a segment of the problem. Students are taught a general problem solving methodology and techniques of library research and creative thinking. They gain experience in planning, questioning, decision-making and produce written and oral reports. The course is primarily for first-year students." (WPI 163)

This description allows for a tremendous leeway in structuring the course; the tenets of the description are that students learn how to solve engineering problems and subsequently write reports that explain their work.

Classroom Structure

ES1020 is taught in a medium-sized multimedia classroom that seats approximately 30 students. Each desk has a computer on it that is capable of running the software necessary for the course. This setup allows the professor to teach students skills regarding engineering software whilst the students at the same time can follow along on the computer.

Typical Class Periods

Generally, the 50-minute class period would be split in half. The first half of the class, the professor would lecture on the engineering topic of the day. During the second half, the professor would introduce the students to one of the engineering software packages that they were to learn. On other occasions, such as those before a project was due, students would simply come to class to work on their project and the professor would make himself available for help.

The ES1020 Student

Although ES1020 is intended to be a class geared towards freshmen, they are not the only audience in the class. There are also a number of upperclassmen, primarily those in non-engineering disciplines, who are simply seeking to gain a general background in engineering. There are also a number of students who are high school seniors that are taking courses at a college level, with ES1020 being an ideal start for them.

Project-Based Curriculum

The deliverables in this class consists of two parts: projects and exams. The projects take approximately two class weeks to complete and combine a number of skills learned in the class.

The exams are also project-oriented and draw from the lessons learned on the projects. The students are also given a number of short problems intended to introduce them to the area of engineering that they would be working with.

Project Topics

The topics covered in the class are intended to be diversified and non-specific to any particular engineering discipline. The topics that were actually assigned covered Engineering Machanics and DC electrical circuit theory. Some example problems that were done in the class are listed below:

Cantilever Beam

In this problem, a beam was firmly attached to a wall and a load was placed on the opposite end. Students were asked to calculate the stress and strain on the beam as well as determine if the beam would be safe enough to support the load. This problem included topics such as equilibrium, stress/strain, and safety factors.

Truck-Body Problem

In this next problem, the object of discussion is a dump truck with a hydraulic piston that raises the dumping bed at a changing angle based on the extension of the piston. Given the amount of force that the piston could exert, students were asked to show the angles at which the dump truck could properly operate. Important topics in this problem include equilibrium, torques, trigonometry, and graphing.

Multiple-Loop DC Circuit

So as to include other engineering disciplines, namely that of electrical engineering, a DC circuit problem was introduced. A large 5-loop circuit with a network of batteries and resistors was given, and students were asked to find the current and voltage at various points. Students were taught the basics of DC circuit theory to complete the project.

Exams

There were 2 exams given in the class and each exam served as a mini-project. These mini-projects would cover the same sort of topics that were covered in the class projects, but would be of a far simpler nature. The exams also make a point of having the students use the various engineering tools to complete different parts of the exam.

Computer-Based Learning

One of the core essentials of this course is that students are taught to use computers in solving engineering problems and creating reports. The specific software that used in this course includes Mathcad, DeltaCAD, and TK Solver.

Mathcad

Mathcad, by MathSoft®, Inc., is a calculation package which includes, according to it's website (www.Mathcad.com), a scratchpad interface, automatic recalculation, and real math notions. Mathcad allows for everything from simple arithmetic calculations to advanced calculus. It includes inline functionality for unit conversion as well as the ability to create and perform calculations on matrices and graphs.

The scratchpad interface is useful because it allows students to type textual elements such as descriptions and explanations directly into the scratchpad and even includes a multitude of formatting tools. Because of this versatile interface, students can generate reports directly in Mathcad without having to use a separate word processing program. With Mathcad's inline unit conversion, students do not have to perform tedious unit conversions manually, which tends to be distracting from the purpose of the calculation. Matrices are useful for solving simultaneous equations, which are done in DC circuit analysis.

One other area that Mathcad excels in is graphing. Using special variables called "range variables," plots can be generated relating two or more variables. This particular piece of functionality is important for generating visuals for engineering reports.

DeltaCAD

DeltaCAD is simply a novice-level Computer-Aided Design (CAD) tool. This package is used to generate various diagrams that students will need in their reports. DeltaCAD does not have a lot of features in and of itself, and drawings made with DeltaCAD should be imported into Mathcad when writing an engineering report.

TK Solver

TK Solver was another package used in the class. TK Solver is simply an equation solver. Students can enter in an equation, define each of the terms in it, enumerate all but one variable and TK Solver will solve for the remaining variable. TK Solver comes packaged with an extensive tutorial that proves to be very easy to use and useful to students.

The Problems with a Traditional Classroom

A traditional classroom varies a great deal from that of the remote electronic classroom. A traditional classroom has flesh-and-blood teachers, but a possible competition for resources and a course flow that is at the same pace for everybody. The eClass is completely electronic, each student has their own resources and can work at a pace that is suitable for their learning style. However, there is no access to real live teachers.

Differing Student Abilities

Perhaps the biggest obstacle in creating a curriculum for a course such as this one is the variety of student abilities in the class. There are essentially two categories of students. The first category are those that are looking for a basic introduction to engineering, but are not interested in pursuing engineering beyond the course. These students typically do not have an engineering mindset and will find the course difficult and will likely have to spend considerable amounts of time learning the skills needed to complete the course. The other type of student is interested in engineering as a profession and is taking the class as a starting point. These students typically have some background in engineering and are likely to have prior computer experience in writing engineering-style reports. These students will be under whelmed by the class and will feel that the class is moving far too slowly. It is these students for which eClass is best suited, as they can finish the course at their own pace, with much less of a need for in-the-flesh instructors. At the same time the electronic medium offers the many possibilities to tailor the class for a particular student groups' needs.

eClass vs. the Traditional Classroom

As we mentioned previously, developing a class for electronic medium involves some challenges that are not present in the traditional classroom. These challenges include access to resources and dealing with the lack of a live teacher.

Access to Resources

One of the advantages of a traditional classroom is that all the materials you need for a course are almost always provided for you. Computers, requisite software, and printing facilities are all typically provided. In a remote environment, students must provide their own computer, must purchase all necessary software, and must provide the means to submit their work, be it by email, fax, or printing and mailing a hard copy. A traditional class, however, often has the problem of scarcity of resources. Sometimes, there will be too few computers, or the computers will have problems (as public lab computers often do). The other scarce resource is access to the instructor, which can be supplanted in an eClass by electronic lessons and lecture videos and access to the Internet.

Teachers vs. Electronic Medium

Since a remote learning environment has no instructor, an alternative must be implemented. An eClass must have all of the materials necessary for a student to complete a particular problem. This can range from textual descriptions of how to complete similar problems to actual videos of an instructor discussing a particular problem or topic. These electronic lessons will probably serve well for many students entering the engineering field. Often, students who feel they have had a bad professor will say, "I learned the course from the book." In an eClass, every student must, in a sense, learn from 'the book.' In today's wired world, students have the opportunity not only to learn from the eClass 'book', but also have the vast body of knowledge in the Internet to assist them.

The Internet

With each passing day, the 'net becomes a more and more viable resource for information. Robust search engines such as Google (http://www.google.com/) and AltaVista (http://www.altavista.com/) have made information easy to find. For instance, if one goes to one of these search engines and search for "Ohm's Law," one will find a plethora of information on the topic. Searching the Internet for information is an extremely valuable skill to learn as an engineering student as well as for an engineer. eClass was designed in such a way as to eliminate the need for Internet to solve the presented problem. However, if a student does feel the need to use the Internet to seek out more information, it is available to them as a resource. We would recommend a student using eClass to obtain Internet access as a prerequisite to starting the course. Internet access obviously becomes a requirement if the instructor chooses to make the courses available over the Internet. This can be done easily and the versatility has been one of the goals of this project.

In addition to research resources, the Internet is a valuable communication tool. An Instructor and his remote student can converse over email, which can start to bridge the distance gap. The instructor can also give the student help with the course just like he would be able to do in person. Finally, the Internet can be used for submission of coursework, avoiding the inconvenience of having to print out and mail coursework.

III. Term 2: Laying the Groundwork

During the second term of the project we focused on analysis and design of different software solutions. We also started exploring the different options for course content production and created a major portion of the contents for the course. In addressing these objectives we also had to take into account the time in with the project needed to be completed as well as the ease of implementation of our solution, both for ourselves and for those developers that would follow us.

Design Objectives

The initial design objective was to create an electronic classroom for students that an instructor can easily customize. While looking at our various options we focused the objectives of the software design to the following three areas: the student, the instructor and the extensibility of the solution.

The Student

The student is the end user of the software product. His or her experiences should be about gaining the knowledge that is presented by the course and the student should not be concerned with the development details of the software. To help ensure positive experience the organization and user interface will be very important.

The Instructor

The instructor could be a traditional classroom instructor or a person designated by a traditional instructor to run an eClass. The instructor takes on the roles of a teacher and of a developer who creates a custom eClass to suit the needs of that particular class. The instructor as we define it will need a certain degree of technical knowledge, but the customization process should be straightforward. This is to ensure that the instructor will concentrate more on content development rather than software development.

Extensibility

The eClass project aims to bring the classroom to more students. The idea of the electronic classroom is a fairly new idea, and not all software solutions will have the same impact. The final product should be implemented such that it will be accessible to a large target audience using different hardware and software platforms.

Design and Implementation

During the design and implementation we explored two different software solutions: a compiled proprietary program using Visual Basic, and an interpreted solution, using HTML. These two solutions have both their strengths and drawbacks as we will discuss in the following sections.

Discarded Solutions

One initial solution involved creating custom software by using the Microsoft Visual Basic programming language. This would have resulted in two pieces of compiled software or software that cannot be modified after the executable files are created. Examples of compiled software include most commercial software such as Microsoft Word and Excel. The two pieces of software would have been the eClass executable program and the eClass Editor program.

eClass Executable Solution

The executable solution presented certain advantages for us in developing the software, but would have created too many drawbacks contrary to our intentions of creating eClass.

Benefits of an Executable Solution

The Microsoft Visual Basic environment provides an easy to use environment for creating simple to highly sophisticated graphical user interface (GUI) applications. This was the main benefit of using Visual Basic over using other programming environments such as Visual C++. By using Visual Basic we would have saved a tremendous amount of time in writing the code for the GUI and also saved a lot of time in creating a workable demo of the software.

Another advantage of creating a custom software solution was that Visual Basic would create a final interface similar to most other Microsoft Windows applications. Also, this solution would work on a wider variety of platforms than an HTML solution would.

Drawbacks of an Executable Solution

As previously mentioned, this discarded solution would have been compiled software. Once we finished the project and made our final changes to it, the next step would have been to create an executable or ".exe" file. Implementing eClass this way meant that the instructor would need to go back to the Visual Basic environment to create his or her own custom eClass. That option would have complicated the process for the instructor and possibly turned away some instructors from adopting eClass.

The second drawback of the executable solution was that it would only function on the Microsoft Windows operating system. To even consider creating a version for a different operating system would have required a great deal of development in a different programming language. The alternative was the HTML solution, which is the solution that was actually implemented.

eClass Editor Solution

An eClass Editor would have made the customization process easier, whether an executable solution or an HTML solution was used. This was not implemented due to the lengthy amount of time that would be required to implement it.

Benefits of an Editor Solution

The main reason for the idea of an eClass editor was that we wanted the instructor to have the ability to create custom an eClass without going to an integrated development environment and having to recompile the entire program. Had we chosen to create an editor for the executable solution it would have been developed such that the editor would write the customization information to a file, which the main program would then read.

The editor could conceivably have also been created for the HTML solution, which we adopted instead of the executable solution. In this case the editor would generate entire HTML files according to a set of options that the instructor selected.

Whether the editor was developed for the executable solution or the HTML solution, it would have dramatically reduced the involvement of the Instructor with the development details providing him or her more assistance and automation in the editing process.

Drawbacks of an Editor Solution

Overall, creating an editor would have been a nice idea; however, the main drawback was the significant amount of time required for the development of the editor and prevented us from devoting the necessary time to the main eClass program and content creation for the "Introduction to Engineering" eClass.

Chosen Solution: HTML (Web page)

The HTML (web page) solution was chosen because it addressed the drawbacks mentioned with the discarded solutions. It also goes further in meeting the three objectives that we set for the design of the eClass software.

Benefits of an HTML solution

HTML is the standard language used for creating web pages that can be viewed both locally on a personal computer and by being accessed any where in the world over the Internet.

Compatibility

eClass is developed as a web page, very much like a web page one would ordinarily come across when browsing the world wide web. The benefit of creating a web page as opposed to a compile program is that web pages are compatible with a variety of different operating systems, thereby giving more people access to eClass.

Versatility

The HTML solution results in a set of files that we have placed on a CD-ROM. All the user has to do is insert the CD-ROM into the drive and the web page will load. The directory and file structure is such that all the files can very easily be copied from the CD-ROM on to an internet sever, making the creating of an internet web page as opposed to a local page seamless.

Ease of Use

A web page has a great advantage over other solutions because most computer users are already familiar with browsing the World Wide Web. This is highly beneficial because it will cut down on the need for training students.

Ease of Development

A web page can be created fairly easily, especially when compared to more complicated Visual Basic or C++ code. Most ordinary web pages that the instructor would need to create can be made using Microsoft Word in much the same way as regular documents are created. However, for the ambitious user a vast array of more advanced HTML development tools are available for adding special effects and database management.

Drawbacks of an HTML Solution

The HTML solution's drawbacks mostly have to do with the choice of browsers. In order to view a web page locally on a personal compute or over the Internet, the user needs software called a web browser. HTML files and web pages are independent of the system because the web browser used to view them is written especially for each different system.

Ideally writing one web page would be enough. However, different browsers support the more advanced features for creating web pages differently. eClass was developed using HTML, but in its present form is only compatible with the Microsoft Internet Explorer 5.5 browser. To make the page compatible with other browsers is not nearly as time consuming as creating multiple compiled software, but that was beyond the aim of this project.

It should be noted that Microsoft Internet Explorer 5.5 browser is freely available from the Microsoft web page, and that Internet Explorer is the most widely used browser. Thus, the aim of making eClass available to as many people as possible (within the scope of this project) was met.

Content Production

Creating this eClass required a fair amount of content production; in addition to making the package, we had to put something in it. There were basically three types of content: text documents, application documents, and videos.

Document Preparation

Many of the lessons we created for eClass were simply text-based lessons, with inline images to explain certain points. Because we are teaching the use of a number of engineering packages, however, we had to create a number of documents that could be viewed with those engineering packages. For example, when we demonstrate a certain point in DeltaCAD, we include the DeltaCAD document so that the student can open it up and observe it.

Videos

Included in this eClass package are a number of videos that explain the various topics covered. There are two distinct types of videos, however, application videos and live videos.

Application Videos

Application videos are animated screen captures of an application being used in real time. These videos include an audio track of the instructor explaining the uses of the particular application. This is particularly useful in that the student can follow along with the video, using the application in a different window as if the instructor were right beside the student.

Live Videos

The other type of video is a live video. This is where an video camera is used to record a live segment of a lecture or some similar demonstration. For example, this particular eClass implementation includes live videos of actual demonstrations that occurred in the course while the product was being prepared.

Space Limitations

This implementation of eClass will be distributed on a standard 650 megabyte CD-ROM. Due to this size constraint, the size and quality of video files used is limited. To get the file sizes down to reasonable amounts, we used video compression to make the files smaller. In the future, when DVD writing is less expensive, a DVD can be made, which can hold as much as 17 gigabytes of information. This would allow for an eClass with much higher quality video and longer vide content than is presently possible.

IV. Term 3: The Final Product

The third and final term of this project was spent creating the final eClass product. This required putting all the content together in a uniformly organized manner as well as creating a consistent user interface.

The Parts of the eClass Software

The eClass package can be broken down three sections: the left-hand navigation pane, the main frame, and the video pop-up lessons.

The Navigation Menu

Morten Wang, a Norwegian JavaScript programmer, originally wrote the tree-style menu seen in the left-hand pane. We have adapted his work to fit our needs, modifying the look and feel of the menu with our own custom colors and content. This hierarchical menu is very similar to the Microsoft® Windows® Explorer, which is why we chose it. Branches of the tree can be expanded or collapsed at will, allowing a user to see only the portion that he/she wishes to see.

This menu was extremely useful because we can organize an eClass into a hierarchical format; each section having it's own root folders, with sub-folders for the project, lessons, etc. For someone wishing to create an eClass, this is the most difficult part to modify; however, by studying the appendix and the files provided with our eClass, it should be fairly straight forward to make modifications.

The Main Frame

The right-hand pane of the window is what we will refer to as the main frame. This is where all of the content (with the exception of the videos) will appear. When a user clicks on an icon in the navigation menu, the actual content associated with that icon will appear in this frame.

Video Lessons

For certain lesson sections, there is a video component. These sections contain a link that will open the video and automatically start playing it. The video player opens in a separate window for both organizational and aesthetic design reasons. This implementation is dependent on features that were added to Internet Explorer 5.5 and thus that version of the browser is necessary for correct operation of the videos.

eClass Structure

The eClass package is broken into a number of sections called "units." Each unit is designed to teach a certain set of skills to the student. The only exception to this is the first unit, which is simply an introduction that contains the expectations of the students and documentation on how to go about completing the course.

Unit Breakdown

A typical unit is divided into 3 main sections. The first section we'll refer to as "Lessons", which contains all instructional materials that go with the unit. This section contains instructional videos as well as HTML-based instructional text. All the methodologies that are needed to solve the problem are included here.

The second section contains practice problems. These problems are similar in nature to the various subsections of the actual project, but are much smaller in scope. Each of these practice problems also has the solution provided so that a student can check his or her work to ensure that they're on the right track.

The third section contains the actual project problem. This section includes not only the problem description but also all the supporting documentation that is needed to complete the project.

The User Interface

In designing a package such as this one, its ease of use is an important factor. The interface should be easy enough to understand that a novice computer user can use the package.

Design for Usability

There are really only two areas where user interface is an issue in this product. The navigation pane uses an explorer-style interface to browse the contents of the eClass. The explorer-style interface should be very intuitive to even basic computer users, as it is implemented throughout the Windows operating system. The header, menu, and main sections were also carefully designed implemented to give clear sense of where different things are located. Attention was also given to making the interface and the user interaction a consistent and visually pleasing experience.

V. eClass: A Student's User Manual

As an eClass student, there are a number of things that should be kept in mind when completing this course. This section is written with an ES1020 student in mind with the caveat that it can be adapted to be tailor-fit to any course.

Overview

An electronic class is very different in nature from a normal brick-and-mortar classroom. For the eClass, all instructional and reference materials have been provided in the package. Also, based on how your instructor arranges the class, the software needed for the course may be provided or must be purchased separately.

It is further expected that you have an Internet connection, for two reasons. The first is that you are expected to communicate your progress to your instructor via email and may use exactly this sort of exchange to ask questions if you get stuck. Additionally, your instructor may or may not have set up the ability for class-wide discussion forums for collaborative discussions of the problems, which would also be accessed via the Internet. Most importantly your instructor may choose to make the eClass available entirely over the Internet instead of on a CD-ROM.

Recommended Flow

All the materials for this eClass are provided on the CD-ROM, however, there is a certain order that should be followed for best results. First, read the project description to get a sense of what is needed to solve the project. Next, go through each of the example problems and attempt to solve them. If you find that you cannot complete a problem, go to the lessons section and view the appropriate lesson that relates to the problem. After completing each problem, check your answer. Once you have successfully completed all the problems, go back to the project description and complete the project.

Deliverables

For this particular course, the only deliverables are the actual project write-ups themselves. These should be submitted in such a manner as designated by the instructor. All other materials provided with the course are for reference only; your grade will be based on the quality of the report that is submitted.

VI. Implementation: An Instructor's Guide

This section is intended as a guide to an instructor that wishes to implement an eClass. There are a number of important things to consider when actually implementing the course for distance learning that may not be inherently obvious. The instructor should provide, in hardcopy form, certain details about the course as discussed below, in addition to contact information for the instructor.

Materials

First, all materials that are needed for this course or instructions on how to obtain them must be included. For example, if a specific book is required for the course, it should be included in the package. If it is more convenient, the student can be provided with the name of an online textbook vendor so that they can order the book themselves. The same arrangement also applies to software needed for a course. If the school has the appropriate license for a specific software title, it can be included without cost to the student. If the school does not have a license, the best method is to pre-obtain all of the software and tack on the charge of it to the student.

Communication

Another important concern in implementing an eClass is the amount of communication that occurs between instructor and student. The instructor should be willing to provide help via email when students encounter difficulties in the course. Additionally, if the course is to be administered to a number of students simultaneously, it is recommended that the professor initiate a class-wide discussion forum so that the students can interact with each other and learn collaboratively. The ideal format for such an interaction is a type of forum known as a "portal."

Portal

A portal is essentially a one-stop shop for the sharing of information. WPI has a campus-wide forum that can be used by faculty and students for distribution and submission of assignments, class discussions, and dissemination of classmate contact information. This portal, called myWPI, has come into widespread use as of the writing of this document.

For implementation of an eClass at WPI, it is recommended that the instructor make use of myWPI for students to discuss problems they encounter as well as for turning in assignments. However, the design and implementation of eClass is such that it can be made available as an independent package if for whatever reason other support pages can not be created.

Testing

The difficulty in distance learning is the facilitation of tests. There are a number of solutions to this, each of which has their potential benefits and drawbacks.

No Testing

The first solution is that testing is not conducted. For project-based courses, this may be an option. The advantage of this solution is that the course can be conducted completely remotely. The drawback to this solution is that there is a large potential for academic dishonesty.

One-Time Testing

Another possible solution is that the instructor designates a specific date, most likely at the end of the academic term, where students in the class are to appear in person to take a final exam. This allows for testing of students with a minimal amount of in-person appearance. Although this essentially eliminates the potential for academic dishonesty, it does force students that are taking the course from long distances to travel to the school to take the exam.

Electronic Time-Limited Testing

Yet another solution is for the instructor to set a specific date and time where an exam would be distributed electronically. The students would then be given a deadline of a few days or a few hours from when the exam is distributed, depending on the course. If testing is an absolute must and the students in the course are widespread throughout the world, this would probably be the best solution. With new software solutions this type of testing is likely to mature and become more practical. With the availability of high-speed Internet connections and secure identification schemes it should be fairly easy to conduct a personal conference between the instructor and the student for the purpose of administering a test.

VII. Conclusion

We are living in the computer age. It will be up to historians to determine if we are well in to it or if the computer age has barely started, but if one thing is clear, it is the dramatic impact that computer technology has already had on our world. In these changing technological times, there are many exciting opportunities that are available for such things as distance learning. Because of innovations like the Internet and high-speed computing, information is available worldwide at the press of a button. These innovations allow the restrictive brick-and-mortar classroom to be expanded to the electronic world. The impact of this expansion has been and will continue to be dramatic as information will be widely available to everyone in a manner that has not been possible before the computer age. As shown in our eClass software, lectures can be converted into electronic handouts and videos, student collaboration can be brought over into Internet-based discussion forums, and the distribution/collection model of the traditional classroom can be converted into electronic submission methods. With a small amount of initial work by an instructor, an eClass can be distributed to students around the world for continual use, year after year. We see eClass as an exciting technology that can serve as both a functional medium for distance learning, and also as an example of the possibilities that are yet to come.

Works Cited

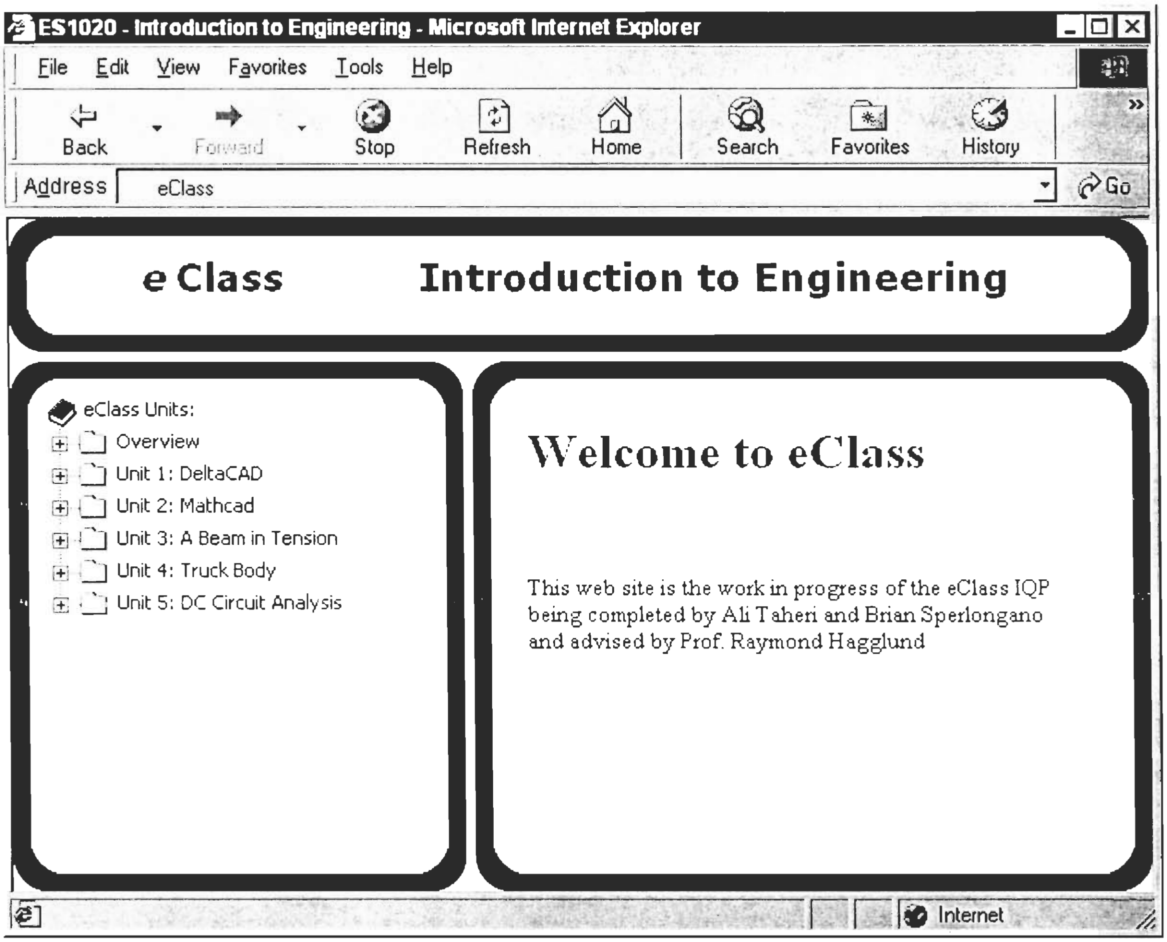
Mathcad.com, http://www.Mathcad.com/. 23 February 2001

Morten's Javascript Tree Menu, http://www.treemenu.com/. 1 March 2001

WPI 2000-2001 Undergraduate Catalog. WPI 2000; WPI; Worcester, MA

Appendix A: Overall Layout

The following screen shot shows the layout of the eClass application. In this image, you can see the title bar at the top, the navigation pane on the left, and the main pane on the right.



Welcome to eClass

This web site is the work in progress of the eClass IQP being completed by Ali Taheri and Brian Sperlongano and advised by Prof. Raymond Hagglund

Appendix B: Root Index File Listing

What follows is a listing of the root index file, which gives the overall look and feel of the site. The extensive hierarchical structure is necessary to maintain the correct overall layout. This file is a available in the root directory of the CD-ROM.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"
      "http://www.w3.org/TR/REC-html40/frame.dtd">
<html>

  <head>
    <title>ES1020 - Introduction to Engineering</title>
  </head>

  <frameset frameborder="0" framespacing="5" border="0" rows="75,100%" id=boo>

    <frameset rows="20,100%,20" framespacing="0" border="0" frameborder="0">
      <frame src="topFrame.html" name="htopFrame" noresize scrolling="NO" target="_self">
      <frameset cols="20,100%,20">
        <frame src="leftFrame.html" name="hleftFrame" noresize scrolling="NO" marginwidth="0"
marginheight="0">
        <frame src="heading.html" name="heading" noresize scrolling="NO">
        <frame src="rightFrame.html" name="hrightFrame" noresize scrolling="NO">
      </frameset>
      <frame src="bottomFrame.html" name="hbottomFrame" noresize scrolling="NO" marginwidth="0"
marginheight="0">
    </frameset>

    <frameset frameborder="NO" framespacing="5" border="0" cols="250,*">

      <frameset rows="0,20,100%,20" framespacing="0" border="0" frameborder="0">
        <frame marginwidth="0" marginheight="0" src="code.html" name="code" noresize scrolling="NO"
frameborder="0">
        <frame src="topFrame.html" name="mtopFrame" noresize scrolling="NO" target="_self">
        <frameset cols="20,100%,20">
          <frame src="leftFrame.html" name="mleftFrame" noresize scrolling="NO" marginwidth="0"
marginheight="0">
          <frame src="menu_empty.html" name="menu" noresize scrolling="auto">
          <frame src="rightFrame.html" name="mrightFrame" noresize scrolling="NO">
        </frameset>
        <frame src="bottomFrame.html" name="mbottomFrame" noresize scrolling="NO" marginwidth="0"
marginheight="0">
      </frameset>

      <frameset rows="20,100%,20" framespacing="0" border="0" frameborder="0">
        <frame src="mainTopFrame.html" name="mainTopFrame" noresize scrolling="NO" target="_self">
        <frameset cols="20,100%,20">
          <frame src="mainLeftFrame.html" name="mainLeftFrame" noresize scrolling="NO"
marginwidth="0" marginheight="0">
          <frame src="main.html" name="text" noresize>
          <frame src="mainRightFrame.html" name="mainRightFrame" noresize scrolling="NO">
        </frameset>
        <frame src="mainBottomFrame.html" name="mainBottomFrame" noresize scrolling="NO"
marginwidth="0" marginheight="0">
      </frameset>

    </frameset>
    </frameset>
</html>
```

Appendix C: eClass Start Pages

Heading HTML file

The following is a listing of the heading.html file, which contains the banner-style animation shown in the topmost frame of the eClass window. The file can contain any type simple to complex HTML that you would like to use for the title bar that will always be present. This file is a available in the root directory of the CD-ROM.

```
<html>

  <head>
    <script language="JavaScript" fptype="dynamicanimation">
<!--
function dynAnimation() {}
function clickSwapImg() {}
//-->
</script>
    <script language="JavaScript1.2" fptype="dynamicanimation" src="animate.js">
</script>
  </head>

  <body onload="dynAnimation()" bgcolor="#f4f4f4" topmargin="0">
    <table border="0" width="100%" cellspacing="0" cellpadding="0">
      <tr>
        <td width="7%">
          <p align="center"></p>
        </td>
        <td width="4%">
          <p align="right" dynamicanimation="fpAnimflyLeftFP1" id="fpAnimflyLeftFP1"
style="position: relative !important; visibility: hidden" language="Javascript1.2"><b><font
face="Verdana" size="5"><i><font color="#000080">e</font></i></font></b></p>
        </td>
        <td width="15%">
          <p dynamicanimation="fpAnimzoomOutFP1" id="fpAnimzoomOutFP1" style="position: relative
!important; visibility: hidden" language="Javascript1.2"><b><font face="Verdana" size="5"><font
color="#0000FF">Class</font><font size="5" color="#0000FF"> </font></font></b></p>
        </td>
        <td width="76%">
          <p align="center" dynamicanimation="fpAnimdropWordFP1" id="fpAnimdropWordFP1"
style="position: relative !important; visibility: hidden" language="Javascript1.2"><b><font
color="#000080" face="Verdana" size="5">Introduction to Engineering</font></b></p>
        </td>
      </tr>
    </table>
  </body>

</html>
```

Heading JavaScript Animation

The actual animation of the text is done in JavaScript. This file is named animate.js and is located in the root of the eClass CD-ROM.

Main Start Page

The contents of the main.html page in the root of the eClass CD-ROM are displayed in the main pane when the eClass is initially opened up. This can be a simple HTML file and should contain some sort of greeting or initial instructions.

Appendix D: Navigation Menu Code

This code is the back end to the navigation pane. It was modified from an original file that can be obtained from http://www.treemenu.com/. This listing can be used as a guide for creating navigation panes for new eClasses. This file is a available in the root directory of the CD-ROM.

```
<html>
<head>
<title>ES1020 - Introduction to Engineering</title>

<script type="text/javascript" src="mtmcode.js">
</script>

<script type="text/javascript">
// Morten's JavaScript Tree Menu
// written by Morten Wang <morten@treemenu.com> (c) 1998-2000
// This is version 2.2.5, dated 2000-03-05

// The script is freely distributable
// It may be used (and modified) as you wish, but retain this message
// For more information about the menu visit its home page
// http://www.treemenu.com/

/******************************************************************************
* User-configurable options.                                                  *
******************************************************************************/

// Menu table width, either a pixel-value (number) or a percentage value.
var MTMTableWidth = "100%";

// Name of the frame where the menu is to appear.
var MTMenuFrame = "menu";

// variable for determining whether a sub-menu always gets a plus-sign
// regardless of whether it holds another sub-menu or not
var MTMSubsGetPlus = true;

// variable that defines whether the menu emulates the behaviour of
// Windows Explorer
var MTMEmulateWE = true;

// Directory of menu images/icons
var MTMenuImageDirectory = "menu-images/";

// Variables for controlling colors in the menu document.
// Regular BODY atttributes as in HTML documents.
var MTMBGColor = "#f4f4f4";
var MTMBackground = "";
var MTMTextColor = "black";

// color for all menu items
var MTMLinkColor = "black";
```

```
// Hover color, when the mouse is over a menu link
var MTMAhoverColor = "darkblue";

// Foreground color for the tracking & clicked submenu item
var MTMTrackColor ="red";
var MTMSubExpandColor = "red";
var MTMSubClosedColor = "red";

// All options regarding the root text and it's icon
var MTMRootIcon = "book.gif";
var MTMenuText = "eClass Units:";
var MTMRootColor = "black";
var MTMRootFont = "Tahoma";
var MTMRootCSSize = "85%";
var MTMRootFontSize = "-1";

// Font for menu items.
var MTMenuFont = "Tahoma";
var MTMenuCSSize = "85%";
var MTMenuFontSize = "-1";

// Variables for style sheet usage
// 'true' means use a linked style sheet.
var MTMLinkedSS = false;
var MTMSSHREF = "style/menu.css";

// Whether you want an open sub-menu to close automagically
// when another sub-menu is opened.  'true' means auto-close
var MTMSubsAutoClose = false;

// This variable controls how long it will take for the menu
// to appear if the tracking code in the content frame has
// failed to display the menu. Number if in tenths of a second
// (1/10) so 10 means "wait 1 second".
var MTMTimeOut = 1;

/******************************************************************************
* User-configurable list of icons.                                            *
******************************************************************************/

var MTMIconList = null;
MTMIconList = new IconList();
MTMIconList.addIcon(new MTMIcon("menu_link_external.gif", "http://", "pre"));
MTMIconList.addIcon(new MTMIcon("menu_link_boo.gif", "http://", "pre"));
MTMIconList.addIcon(new MTMIcon("menu_link_pdf.gif", ".pdf", "post"));
MTMIconList.addIcon(new MTMIcon("menu_link_dc.gif", ".dc", "post"));
MTMIconList.addIcon(new MTMIcon("menu_link_mc.gif", ".mcd", "post"));

/******************************************************************************
* User-configurable menu.                                                     *
******************************************************************************/

// Main menu.
var menu = null;
menu = new MTMenu();
menu.MTMAddItem(new MTMenuItem("Overview"));
menu.MTMAddItem(new MTMenuItem("Unit 1: DeltaCAD"));    //0
menu.MTMAddItem(new MTMenuItem("Unit 2: Mathcad"));     //1
menu.MTMAddItem(new MTMenuItem("Unit 3: A Beam in Tension")); //2
menu.MTMAddItem(new MTMenuItem("Unit 4: Truck Body ")); //3
menu.MTMAddItem(new MTMenuItem("Unit 5: DC Circuit Analysis"));  //4

var overview = null;
overview = new MTMenu
overview.MTMAddItem(new MTMenuItem("Instructions", "general/navigate_eClass.html", "text"));
overview.MTMAddItem(new MTMenuItem("Report Format", "general/report_format.html", "text"));
menu.items[0].MTMakeSubmenu(overview);

var unit1 = null;
unit1 = new MTMenu
unit1.MTMAddItem(new MTMenuItem("Lessons"));
```

```
menu.items[1].MTMakeSubmenu(unit1);

 var u1Lessons = null;
 u1Lessons = new MTMenu
 u1Lessons.MTMAddItem(new MTMenuItem("DeltaCAD Environment", "video/deltacad_environment.html",
"text"));
 u1Lessons.MTMAddItem(new MTMenuItem("DeltaCAD Menus", "video/deltacad_menus.html", "text"));
 u1Lessons.MTMAddItem(new MTMenuItem("DeltaCAD Grid", "video/deltacad_saving_grid.html",
"text"));
 u1Lessons.MTMAddItem(new MTMenuItem("DeltaCAD Tabs", "video/deltacad_tabs.html", "text"));
 u1Lessons.MTMAddItem(new MTMenuItem("Shapes and Lines", "video/deltacad_car_1.html", "text"));
 u1Lessons.MTMAddItem(new MTMenuItem("Parallel lines", "video/deltacad_car_2.html", "text"));
 u1Lessons.MTMAddItem(new MTMenuItem("Corner Tool & Trimming", "video/deltacad_car_3.html",
"text"));
 u1Lessons.MTMAddItem(new MTMenuItem("Arc Tool", "video/deltacad_car_4.html", "text"));
 u1Lessons.MTMAddItem(new MTMenuItem("Grid & Snap", "video/deltacad_diagram_1.html", "text"));
 u1Lessons.MTMAddItem(new MTMenuItem("Arrows and Labels", "video/deltacad_diagram_2.html",
"text"));
 u1Lessons.MTMAddItem(new MTMenuItem("Dimensions & Positioning",
"video/deltacad_diagram_3.html", "text"));
 unit1.items[0].MTMakeSubmenu(u1Lessons);

var unit2 = null;
unit2 = new MTMenu
unit2.MTMAddItem(new MTMenuItem("Lessons"));
menu.items[2].MTMakeSubmenu(unit2);

  var u1Lessons = null;
  u2Lessons = new MTMenu
  u2Lessons.MTMAddItem(new MTMenuItem("Mathcad Basics", "video/Mathcad_basics.html", "text"));
  u2Lessons.MTMAddItem(new MTMenuItem("Range Variables & Graphs",
"video/Mathcad_graph_rangevars.html", "text"));
  unit2.items[0].MTMakeSubmenu(u2Lessons);

var unit3 = null;
unit3 = new MTMenu
unit3.MTMAddItem(new MTMenuItem("Lessons"));
unit3.MTMAddItem(new MTMenuItem("Practice"));
unit3.MTMAddItem(new MTMenuItem("Solutions"));
unit3.MTMAddItem(new MTMenuItem("Files"));
unit3.MTMAddItem(new MTMenuItem("Project"));
menu.items[3].MTMakeSubmenu(unit3);

  var u3Lessons = null;
  u3Lessons = new MTMenu

  u3Lessons.MTMAddItem(new MTMenuItem("Basic Quantities", "video/basic_quantities.html",
"text"));
  u3Lessons.MTMAddItem(new MTMenuItem("Free Body Diagrams", "video/freebody_diagram_seesaw.html",
"text"));
  u3Lessons.MTMAddItem(new MTMenuItem("Newton's Laws", "video/newton_laws_motion.html", "text"));
  u3Lessons.MTMAddItem(new MTMenuItem("Particles", "video/particle_rigidbody_models.avi",
"text"));
  u3Lessons.MTMAddItem(new MTMenuItem("Equilibrium Conditions",
"video/equilibrium_conditions.html", "text"));
  u3Lessons.MTMAddItem(new MTMenuItem("Torque", "video/equilibrium_torques.html", "text"));
  u3Lessons.MTMAddItem(new MTMenuItem("Couple Moments", "video/couple_moment.html", "text"));
  u3Lessons.MTMAddItem(new MTMenuItem("Linear Elasticity", "video/linear_elasticity.html",
"text"));
  u3Lessons.MTMAddItem(new MTMenuItem("Stress", "video/stress.html", "text"));
  u3Lessons.MTMAddItem(new MTMenuItem("Strain", "video/strain_deformation.html", "text"));
  u3Lessons.MTMAddItem(new MTMenuItem("Stress vs. Strain", "video/stress_strain_curve.html",
"text"));
  u3Lessons.MTMAddItem(new MTMenuItem("Safety Factor", "video/safety_factor.html", "text"));

  unit3.items[0].MTMakeSubmenu(u3Lessons);

  var u3Practice = null;
  u3Practice = new MTMenu
  u3Practice.MTMAddItem(new MTMenuItem("Free-Body Diagram", "unit3/see-saw-fbd-example.html",
"text"));
```

```
  u3Practice.MTMAddItem(new MTMenuItem("Equilibrium", "unit3/equilibrium_problems.html",
"text"));
  u3Practice.MTMAddItem(new MTMenuItem("More Equilibrium", "unit3/man-beam-equilibrium-
example.htm", "text"));
  unit3.items[1].MTMakeSubmenu(u3Practice);

  var u3Solutions = null;
  u3Solutions = new MTMenu
  u3Solutions.MTMAddItem(new MTMenuItem("Free-Body Diagram #1", "unit3/see-saw-fbd-example-
solution.html", "text"));
  u3Solutions.MTMAddItem(new MTMenuItem("Equilibrium", "unit3/equilibrium_solutions.html",
"text"));
  unit3.items[2].MTMakeSubmenu(u3Solutions);

  var u3Files = null;
  u3Files = new MTMenu
  u3Files.MTMAddItem(new MTMenuItem("Read First", "general/files_readme.html", "text"));
  u3Files.MTMAddItem(new MTMenuItem("Free-Body Diagram #1 (DC)", "unit3/see-saw-fbd-example.dc",
"text"));
  u3Files.MTMAddItem(new MTMenuItem("Free-Body Diagram #2 (DC)", "unit3/plank-fbd-example.dc",
"text"));
  u3Files.MTMAddItem(new MTMenuItem("More Equilibrium #1 (MC)", "unit3/man-beam-equilibrium-
example.mcd", "text"));
  u3Files.MTMAddItem(new MTMenuItem("More Equilibrium #2 (MC)", "unit3/doorframe-equilibrium-
example.mcd", "text"));
  unit3.items[3].MTMakeSubmenu(u3Files);

  var u3Project = null;
  u3Project = new MTMenu
  u3Project.MTMAddItem(new MTMenuItem("Description", "unit3/project1/project1.htm", "text"));
  u3Project.MTMAddItem(new MTMenuItem("Instructions", "unit3/project1/project1-howto.htm",
"text"));
  unit3.items[4].MTMakeSubmenu(u3Project);

var unit4 = null;
unit4 = new MTMenu
unit4.MTMAddItem(new MTMenuItem("Lessons"));
unit4.MTMAddItem(new MTMenuItem("Practice"));
unit4.MTMAddItem(new MTMenuItem("Solutions"));
unit4.MTMAddItem(new MTMenuItem("Files"));
unit4.MTMAddItem(new MTMenuItem("Project"));
menu.items[4].MTMakeSubmenu(unit4);

  var u4Lessons = null;
  u4Lessons = new MTMenu
  u4Lessons.MTMAddItem(new MTMenuItem("Free Body Diagrams 2",
"video/freebody_diagram_truckramp.html", "text"));
  unit4.items[0].MTMakeSubmenu(u4Lessons);

  var u4Practice = null;
  u4Practice = new MTMenu
  u4Practice.MTMAddItem(new MTMenuItem("Force Components", "unit4/practice1.html", "text"));
  u4Practice.MTMAddItem(new MTMenuItem("Solving with Ranges", "unit4/practice2.html", "text"));
  u4Practice.MTMAddItem(new MTMenuItem("Plotting", "unit4/practice3.html", "text"));
  unit4.items[1].MTMakeSubmenu(u4Practice);

  var u4Solution = null;
  u4Solution = new MTMenu
  u4Solution.MTMAddItem(new MTMenuItem("Force Components", "unit4/solution1.html", "text"));
  u4Solution.MTMAddItem(new MTMenuItem("Solving with Ranges", "unit4/solution2.html", "text"));
  u4Solution.MTMAddItem(new MTMenuItem("Plotting", "unit4/solution3.html", "text"));
  unit4.items[2].MTMakeSubmenu(u4Solution);

  var u4Files = null;
  u4Files = new MTMenu
  u4Files.MTMAddItem(new MTMenuItem("Read First", "general/files_readme.html", "text"));
  u4Files.MTMAddItem(new MTMenuItem("Solving with Ranges", "unit4/solution2.mcd", "text"));
  u4Files.MTMAddItem(new MTMenuItem("Plotting", "unit4/solution3.mcd", "text"));
  unit4.items[3].MTMakeSubmenu(u4Files);

  var u4Project = null;
  u4Project = new MTMenu
  u4Project.MTMAddItem(new MTMenuItem("Description", "unit4/description.html", "text"));
  u4Project.MTMAddItem(new MTMenuItem("Hints", "unit4/hints.html", "text"));
  unit4.items[4].MTMakeSubmenu(u4Project);

var unit5 = null;
unit5 = new MTMenu
unit5.MTMAddItem(new MTMenuItem("Lessons"));
unit5.MTMAddItem(new MTMenuItem("Practice"));
unit5.MTMAddItem(new MTMenuItem("Solutions"));
unit5.MTMAddItem(new MTMenuItem("Project"));
menu.items[5].MTMakeSubmenu(unit5);

  var u5Lessons = null;
  u5Lessons = new MTMenu
  u5Lessons.MTMAddItem(new MTMenuItem("Electricity Basics", "video/electricity_basics.html",
"text"));
  u5Lessons.MTMAddItem(new MTMenuItem("Kirchovff's Current Law",
"unit5/Lessons/Kirchovff_Current_Law.htm", "text"));
  u5Lessons.MTMAddItem(new MTMenuItem("Kirchovff's Voltage Law",
"unit5/Lessons/Kirchovff_Voltage_Law.htm", "text"));
  u5Lessons.MTMAddItem(new MTMenuItem("Multiple-Loop Problems",
"unit5/Lessons/multiple_loop.htm", "text"));
  unit5.items[0].MTMakeSubmenu(u5Lessons);

  var u5Practice = null;
  u5Practice = new MTMenu
  u5Practice.MTMAddItem(new MTMenuItem("Voltage"));
  u5Practice.MTMAddItem(new MTMenuItem("Resistance"));
  u5Practice.MTMAddItem(new MTMenuItem("Current"));
  unit5.items[1].MTMakeSubmenu(u5Practice);

            var u5PracticeVoltage = null;
            u5PracticeVoltage = new MTMenu
            u5PracticeVoltage.MTMAddItem(new MTMenuItem("Voltage #1",
"unit5/practice/voltage1.html", "text"));
            u5PracticeVoltage.MTMAddItem(new MTMenuItem("Voltage #2",
"unit5/practice/voltage2.html", "text"));
            u5Practice.items[0].MTMakeSubmenu(u5PracticeVoltage);

            var u5PracticeResistance = null;
            u5PracticeResistance = new MTMenu
            u5PracticeResistance.MTMAddItem(new MTMenuItem("Resistance",
"unit5/practice/resistance1.html", "text"));
            u5Practice.items[1].MTMakeSubmenu(u5PracticeResistance);

            var u5PracticeCurrent = null;
            u5PracticeCurrent = new MTMenu
            u5PracticeCurrent.MTMAddItem(new MTMenuItem("Current #1",
"unit5/practice/current1.html", "text"));
            u5PracticeCurrent.MTMAddItem(new MTMenuItem("Current #2",
"unit5/practice/current2.html", "text"));
            u5PracticeCurrent.MTMAddItem(new MTMenuItem("Current #3",
"unit5/practice/current3.html", "text"));
            u5Practice.items[2].MTMakeSubmenu(u5PracticeCurrent);

  var u5Solutions = null;
  u5Solutions = new MTMenu
  u5Solutions.MTMAddItem(new MTMenuItem("Voltage"));
  u5Solutions.MTMAddItem(new MTMenuItem("Resistance"));
  u5Solutions.MTMAddItem(new MTMenuItem("Current"));
  unit5.items[2].MTMakeSubmenu(u5Solutions);

            var u5SolutionsVoltage = null;
            u5SolutionsVoltage = new MTMenu
            u5SolutionsVoltage.MTMAddItem(new MTMenuItem("Voltage #1",
"unit5/solutions/voltage1.html", "text"));
            u5SolutionsVoltage.MTMAddItem(new MTMenuItem("Voltage #2",
"unit5/solutions/voltage2.html", "text"));
            u5Solutions.items[0].MTMakeSubmenu(u5SolutionsVoltage);
            var u5SolutionsResistance = null;
            u5SolutionsResistance = new MTMenu
            u5SolutionsResistance.MTMAddItem(new MTMenuItem("Resistance",
"unit5/solutions/resistance1.html", "text"));
            u5Solutions.items[1].MTMakeSubmenu(u5SolutionsResistance);

            var u5SolutionsCurrent = null;
            u5SolutionsCurrent = new MTMenu
            u5SolutionsCurrent.MTMAddItem(new MTMenuItem("Current #1",
"unit5/solutions/current1.html", "text"));
            u5SolutionsCurrent.MTMAddItem(new MTMenuItem("Current #2",
"unit5/solutions/current2.html", "text"));
            u5SolutionsCurrent.MTMAddItem(new MTMenuItem("Current #3",
"unit5/solutions/current3.html", "text"));
            u5Solutions.items[2].MTMakeSubmenu(u5SolutionsCurrent);

  var u5Project = null;
  u5Project = new MTMenu
  u5Project.MTMAddItem(new MTMenuItem("Description", "unit5/circuit_project.html", "text"));
  unit5.items[3].MTMakeSubmenu(u5Project);

</script>
</head>
```

Appendix E: Video Start Page Template

The following code is a template for the start page for a video file. The purpose of this file is to be an intermediary between the eClass web page and the actual multimedia content. This file must be edited for each video file that you wish to include in an eClass. Each of the bracketed areas must be edited. This file is a available in the video directory of the CD-ROM.

```
<html>
<head>
<title>[Lesson Name]</title>
<script language="JavaScript">
  //Specify the following information
  fileToPlay = "filename.avi";
  fileSize = "1.23";
  fileLength = "1 Min 23 Sec";
  fileWidth  = 123;
  fileHeight = 123;

  //Do not modify!!
  widthOffset   = 90;
  heightOffset  = 90;
  wWidth     = fileWidth + widthOffset;
  wHeight    = fileHeight + heightOffset;
  var i = 0;
  function LoadVideoPlayer(){
   mywin =
window.open("eCVPlayer.html","win","width="+wWidth+",height="+wHeight+"top=0,left=0","toolbar=yes
,resizable=yes")·
   //wait 5 seconds before passing variable
   wait(5);
   mywin.SetVideoFileName(fileToPlay);

  }
  function wait(seconds) {
   now = new Date();
   nowTime = now.getSeconds() ;
   waitTime = nowTime + seconds;
   while (nowTime < waitTime) {
     now = new Date();
     nowTime = now.getSeconds();
   }
  }
 </script>
 </head>

 <body topmargin=0 leftmargin=0>
 <P align=center><B>[Lesson Name Goes Here]</B></P>
 <HR>

 <P><b>Description:</b></P>

 <P>[Lesson description goes here.]</P>

 <P>Click <A href="javascript:LoadVideoPlayer();">here</A> to Load Video Player.</P>
 <P>&nbsp;</P>

 <TABLE cellSpacing=0 cellPadding=0 width=300 align=center border=0>

   <TR>
     <TD><b>File Name:</b> </TD>
```

```
    <TD>
      <center>
      <INPUT style="WIDTH: 180px; BORDER-TOP-STYLE: none; BORDER-RIGHT-STYLE: none; BORDER-LEFT-
STYLE: none; HEIGHT: 22px; BORDER-BOTTOM-STYLE: none" size=28 name=fnametxt disabled>
      </center>
    </TD>
  </TR>
  <TR>
    <TD><b>File Size:</b></TD>
    <TD>
      <center>
      <INPUT style="WIDTH: 180px; BORDER-TOP-STYLE: none; BORDER-RIGHT-STYLE: none; BORDER-LEFT-
STYLE: none; HEIGHT: 22px; BORDER-BOTTOM-STYLE: none" size=27 name=fsizetxt disabled>
      </center>
     </TD>
  </TR>
  <TR>
    <TD><b>Length:</b></TD>
    <TD>
      <center>
      <INPUT style="WIDTH: 180px; BORDER-TOP-STYLE: none; BORDER-RIGHT-STYLE: none; BORDER-LEFT-
STYLE: none; HEIGHT: 22px; BORDER-BOTTOM-STYLE: none" size=27 name=lengthtxt disabled>
     </center>
    </TD>
  </TR>
</TABLE>
<script language=javascript>
   fnametxt.value = fileToPlay;
   fsizetxt.value = fileSize;
   lengthtxt.value= fileLength;
</script>
</body>

</html>
```

Appendix F: Video Player Code

The following code controls the operation of the eClass video player. This file is a available in the video directory of the CD-ROM.

```
<script language="JavaScript">

  function SetVideoFileName(theValue) {
   myVideo.filename = theValue;
    }

  function OnInfo(){
   window.open("help.html","Help","width=300,height=275","toolbar=yes,resizable=yes");
  }

  function OnForward() {
   var curPosition = myVideo.CurrentPosition;
   newPosition = curPosition + 2.0;
   myVideo.CurrentPosition = newPosition;
  }

  function OnRewind() {
   var curPosition = myVideo.CurrentPosition;
   newPosition = curPosition - 2.0;
   if (newPosition < myVideo.SelectionStart)
     newPosition = 0.0;
   myVideo.CurrentPosition = newPosition;
  }

  function OnPlay() {
   myVideo.Volume = 0;
   myVideo.Run();
   myVideo.FullScreenMode = false;
  }

  function OnPause() {
   if (myVideo.CurrentState == 1)
     myVideo.Run();
   else
     myVideo.Pause();
  }

  function OnFScreen() {
   myVideo.FullScreenMode = true;
   OnPause();
   OnPlay();
  }

  function OnStop() {
   myVideo.stop();
  }
  </script>
```

Appendix G: Developing A New eClass

This section is intended to guide an eClass developer in designing a new eClass for production

Course Layout Design

The first step in creating an eClass is to properly design the curricula structure that you will be using. For example, a course may be divided into a number of sections, each with it's own lessons and projects. A course may also be divided into a more hierarchical format with sections, subsections, sub-subsections, and so forth. This is analogous to having a textbook with a hierarchical layout.

Create Content

The next step is to actually create the content that will be used for the course. This may include videotaping lectures, text lessons, handouts, scanned documents, or a plethora of other document types. Although the simplest eClass can be created using HTML files generated by a program such as MS-Word, more savvy developers can implement an array of different content types.

Edit Navigation Pane

Using Appendix D as an example, create a code.html file that generates the appropriate hierarchical structure for your course. The structure of the script is such that each item in the tree has a parent-child relationship, and is statically numbered. Study the code carefully to gain a complete understanding of how it works, and visit http://www.treemenu.com/ for more information.

Edit Look And Feel

Be sure to change the heading banner and main start pages that are course-specific. These files can be found in the eClass root. Also, you may wish to edit the border images to create a look that is appropriate to your particular course. For those interested further information about changing the look and feel of the menu is available at http://www.treemenu.com/