Teaching IE's About the Web

Louis J. Plebani, Joseph C. Hartman
Lehigh University

Overview

Because web applications continue to become increasingly more important in industry solutions, its axiomatic that IE graduates have an understanding of how "The Web" works. A difficulty is that there are numerous, and ever increasing numbers, of technologies that form the components of Web applications. However, the up side of it all is that many of the technologies are simply alternate ways of accomplishing the same thing. This means that if the students achieve a modest level of understanding of an appropriate core set web of application components, they achieve a modest level of understanding of all the technologies.

However, it is still true that once a core set of technologies is selected, the volume of material to be covered can still easily (and most assuredly) overwhelm a semester course if that course attempts to completely cover all the topics in a traditional lecture mode manner. One only has to visit the computer section of the local bookstore and observe the rows of 2-3 inch thick books on each of the technologies. If each of the selected technologies were covered in depth several semesters would be involved. In addition, this would be somewhat of an overkill in relation to the goal of providing a basic understanding of the web.

We developed a course, which used an active learning approach, to attack the above problems. In this course, students learn about web applications by building a web application as a semester project. In order to be successful, it is necessary that the students be motivated to investigate, discover, synthesize, and improvise while doing their project. For this to occur, students must be well acquainted with the intricacies of the problem to be solved by their application. Thus the choice of application, i.e., the problem to be solved, is critical.

Topics Covered

The course addressed those topics that would be involved in the design of most any web site of reasonable complexity. The topics included were HTML, Cascading Style Sheets, Client Side Scripting, Web Servers and Server Side Processing, and Database.

HTML (Hypertext Markup Language): A minimum of time was devoted to actually discussing HTML in class. There were two reasons for this. The first is that most students have some general knowledge of HTML just by virtue of their daily interactions with the web. The second reason is that there is a wealth of tutorials available on the web, which students could access. Students were told to work through a web tutorial that covered basic HTML and that a quiz on HTML basics would be given on a specified date. At the time of the class offering, the tutorial(s) at http://www.w3schools.com were used. During the time from assignment to the scheduled quiz, class time was devoted to discussion of the World Wide Web, how the web works, how browsers fetch and display pages, and the various standards set by the World Wide Web Consortium. As for all the topics discussed in the course, the Internet accessible resources were used as the textbook.

CSS (Cascading Style Sheets): After basic HTML, advanced HTML discussion concentrated on styles. Styles define how HTML elements are displayed and are normally saved in files external to HTML documents. CSS was a breakthrough in Web design because it allows developers to control the style and layout of multiple Web pages all at once. A Web developer can define a style for each HTML element and apply it to as many Web pages as desired. To make a global change, a change is made to style, and all elements in the appropriate web application are updated automatically. As in HTML, time did not permit the detailed discussion of the entire CSS standard. Since, the goal was to introduce the students to the capabilities of styles and not make experts, a problem based learning approach was taken. As in HTML, students were told to work through a web tutorial and a short quiz on basic material would be given. In addition, an assignment was given for the students to create a HTML application that used cascading style sheets. To encourage, the students to investigate as wide as scope as possible, they were told that bonus points would be given to the ugliest application, i.e., the one that misused style sheets the most. This proved effective.

Client Side Scripting: JavaScript was used as the language for client side scripting. This choice was fairly obvious as JavaScript works in all major browsers that are version 3.0 or higher. Furthermore it is a “light” programming language that all students were readily able to understand and use. Client side scripting also involved the study and discussion of Dynamic HTML which is the art of making HTML pages dynamic by using scripting to manipulate the style, layout and contents of the document. The learning approach take was similar to that taken for Cascading Style Sheets. Students were told to work through web tutorials on JavaScript and DHTML. While students were doing the tutorials, class discussion revolved around questions brought to class by puzzled students. A project was introduced which also served as a catalyst for class discussion. The project required the students to develop a game of their choosing. The game would use web technologies discussed in class to date. Complexity from a gaming point of view was irrelevant. The only requirements were that the game have both text and mouse inputs. Students were allowed to work in teams. They were told that evaluation of their assignment would include evaluation by their peers. After the due date, all students in the class were told to assess the games developed by their fellow students. They were required to submit an evaluation that included an overall evaluation of the application and the results of their success in “breaking” (crashing) the application.

Web Servers and Server Side Processing: There are many possibilities that could be used as examples of web servers and server side processing. Because they are representative of these technologies and also fairly ubiquitous in the real world, the course focused on the Apache Web server and PHP server side scripting. Students were encouraged to install Apache with PHP capability on the own computers so that they could develop applications locally. It is important to note that one need not be connected to the Internet to run a web server. A standalone machine can be used for web development and then a completed project uploaded to a “real” web server upon project completion. This is the model that was used for the course. A course web server was also established. This server was hosted on a PC running in the instructor’s office. Packages for download and installation for Apache and PHP are available from http://www.apache.org/ and http://www.php.net/ respectively. As was done for the previous topics, students were instructed to access web tutorials. At this time, the final project was introduced which is discussed below. This project generated most of the class discussion for the rest of the course and regarding server side topics in particular.

Database: The students in the course had all been exposed to database concepts in a previous course so it was not necessary to spend a much time on database concepts. A brief review of SQL was done. SQL is an ANSI (American National Standards Institute) standard for accessing database systems. SQL statements are used to retrieve and update data in a database. The database used for the course was MySQL, which is available for download from http://www.mysql.com. MySQL was chosen because it is the world's most popular Open Source Database. Students were encouraged to download and install MySQL on their local machines.

Final Project

Independently of the course, we have been working on using the web for problem based (active) learning with emphasis on probability, statistics, and engineering economy applications. Our experiences in developing these applications for problem based learning, convinced us that such projects would be "perfect" for the students to work on in the course. Since mostly seniors take the course, the one area that the students have in common is "what it takes to learn". Based on this, the requirement for the course is that each student group develop a problem based learning application which emphasizes the interpretation, data collection, analysis, and decision making around a virtual reality problem which is created as part of their application. The applications were required to have the following characteristics.

- The system was to consist of deterministic parts such as a problem story and parameter generation rules that detail the manner in which specific problem parameters are to be randomly generated.

- The database was to contain a list of student names and passwords.

- When a student logins in and requests a problem the system creates and initializes a unique problem by randomly generating appropriate problem parameters and combining them with the deterministic part of the problem. All generated parameters for each problem were to be stored in a database for future interactions with the student and for solution evaluation.

- The problem specific parameters also were to define an experimentation environment through which the student may query the system to obtain data for analysis. Students were to be allowed to login in/out any number of times to view or request more data for analysis before submitting a solution.

- Upon submission of a solution, the system was to evaluate the solution. Both solution accuracy as well as the resources used in achieving the solution, e.g., the number of data points requested, calendar time, etc., were allowed to influence the final score.

- The system database was to store a history of all student interactions and the results of problem evaluations so that the database could be queried for statistics regarding all problem solvers.

Students were encouraged to develop their applications on their own local machines and upon completion, FTP them to the class server. As was done in the previous project, all students in the class were required to accesses the applications of all other students and submit an evaluation to the instructor.

Student Response

Formal course evaluations are not available at the time of this writing. Anecdotally, based on informal conversations and the instructor estimate of student enthusiasm, students strongly supported the problem based approach to the course.

LOUIS J. PLEBANI
Louis J. Plebani is an Associate Professor of Industrial and Manufacturing Systems Engineering at Lehigh University. He received his Ph.D. in Industrial Engineering at Lehigh, after receiving his M.S. in Operations Research at American University and B.S. in Engineering Physics at Lehigh. He is a member of ASEE and a Registered Professional Engineer in Pennsylvania.

JOSEPH C. HARTMAN
Joseph C. Hartman is an Assistant Professor of Industrial and Manufacturing Systems Engineering at Lehigh University. He received his Ph.D. and M.S. in Industrial Engineering from the Georgia Institute of Technology and a B.S. in General Engineering from the University of Illinois at Urbana-Champaign. He is a member of ASEE, IIE, INFORMS and NSPE. He currently serves as an area editor for *The Engineering Economist*.