Web-Based Learning Environment for a Communications Module

WALEEED AL-NUAIMY, JINGHUA ZHANG, ALAN NOBLE

Department of Electrical Engineering and Electronics, University of Liverpool, Brownlow Hill, Liverpool L69 3GJ, UK

Received 8 January 2001; accepted 26 February 2001

ABSTRACT: Teaching and learning in higher education has recently been transformed by the development and affordability of fast microcomputers and the accessibility of the World Wide Web. The potential to utilise the Web to enhance students’ learning has been recognised, and to this end a pilot web-based system has been developed as an online interactive resource for the teaching and learning of an undergraduate module on Communications within the Department of Electrical Engineering and Electronics. The system comprises course note material, interactive tutorial demonstrations, an interactive self-assessment system, and a facility for asynchronous group interaction. These materials are intended to complement the traditional taught element of the module, and will support new student-centred delivery methods being implemented within the University of Liverpool. © 2001 John Wiley & Sons, Inc.

Keywords: World Wide Web; interactive learning; communications; education

INTRODUCTION

The World Wide Web has immense potential in making information more accessible, more dramatically customisable, and visually more appealing to the user. The development of the Web and associated technologies has had an impact on curriculum development and approaches to teaching and learning. Traditionally, learners have been passive receivers except in performance evaluations in the form of homework, projects, and tests. The current availability of low-cost, high volume, and online access to vast information banks using personal computers has altered the role of the instructor from that of a knowledge source to that of a learning facilitator.

Computer-based learning allows learners to work at their own pace and gain instant feedback while involving themselves in their own learning process. The benefits to the lecturer might include a reduction in the time spent marking assessments, fewer routine enquiries to deal with, and an opportunity to concentrate on other aspects of their work.

Web-based learning is an innovative approach to asynchronous learning in which computer-based training is transformed by the technologies and methodologies of the Web [1]. Delivering courseware over the Web has several advantages over computer-based training methods, including:

- the software (internet browsers) is free and readily available,
- inherent multi-platform capabilities,
The aim was to develop and produce a pilot system serving as an online resource for the teaching and learning of an undergraduate module on Communications within the Department of Electrical Engineering and Electronics. These materials are intended to complement the traditional taught element of this module, and will support new student-centred delivery methods being implemented within the University of Liverpool.

In order to provide a quality online educational experience, it is necessary not only to provide access to the information, but also to provide opportunities for interaction and collaboration, a clear statement of goals and tasks, and some form of assessment process [2]. With these considerations in mind, the system was thus designed to allow learners access to the following virtual classroom spaces:

- online course notes and module syllabus,
- interactive demonstration system,
- self-assessment system,
- a discussion group for asynchronous group communications,
- direct email access to instructor, and
- links to related web sites and external resources.

Figure 1  Basic structure of web-based learning system.
textbooks leave off, and it is hoped this approach will have a greater impact than that which single-dimensional books or static web pages can accomplish.

IMPLEMENTATION APPROACHES

The two most common ways of adding interactivity to Web pages are Common Gateway Interface (CGI) programs and Java applets; these run at opposite ends of the communications link.

The Java language is interpreted by the end user’s Web browser. Although originally intended for animation, they are also very useful for anything that requires screen updates independent of the communications link to the server. Any demonstration or tutorial program written as a Java applet would initially be downloaded from the server, and would then run locally without needing to connect to the server.

CGI programs and forms allow a program and its data at the server end to be shared by a large number of users in the field without needing to distribute client applications other than the browser. The main processing program of a CGI-based demo would run on the server and user interfaces would be provided on the Web only to collect input information. Such an implementation would generally require greater bandwidth than a Java implementation.

For this engineering course on communications, engineering calculations and representations in the interactive demonstrations are necessary. Since Java is a general-purpose programming language, not a scientific or engineering tool, it has no functions for modulation, demodulation, filtering, or spectrum representation. If Java applets were used to implement these communications tutorials, the programming would be complex for the purpose of interface design, communication-related functions, engineering calculations, plotting effects, and so on. Purpose-written code would be required to implement even the most basic functions, resulting in inefficient code.

With CGI on the other hand, the choice of programming language is open to the developer. One of the best-suited engineering programming language is Matlab™, as there are extensive libraries of special-purpose engineering functions available within the university. As the graphical user input interfaces would be produced by the browser from hypertext, and the calculations and plotting effects can be generated with little effort using Matlab, considerably less effort would be required to produce several high-quality tutorials.

With all the above consideration in mind, the present system was implemented using Matlab Web Server. This combines most of the advantages mentioned above at the expense of a slight decrease in responsiveness.

Matlab Web Server

Through the use of standard HTML documents and forms, Matlab Web Server allows the development of Web-deployable applications from standard Matlab components. The HTML documents serve as a point-and-click interface for the application being deployed. As a result, users of the application are not required to learn Matlab, and Matlab need not be running locally on the client machine. As Figure 1 shows, the client browser only sends variables via an HTML form to the server, which processes these input variables and sends back output variables.

The process of creating Web server application involves the creation of:

1. An HTML input document for data submission to the server. This would typically include a form with various user settings, a figure or diagram of some sort, and some instructions (such as that in Figure 2).
2. An HTML output document for display of the output computations. This would be in the form of a plot or other graphic and numeric representation.
3. A Matlab M-file to process input data and compute results.
4. A test file to validate code before distributing the application over the Web.

A set of 11 interactive tutorial demonstrations was constructed to cover the main topics of this communications module. Most of these were concerned with aspects of signal modulation and demodulation, so all the programs are similar in nature, and little effort was needed to produce the effects shown in Figure 3.

SELF-ASSESSMENT

Our objective here was to provide the learners with an area where they can test what they have learnt about a topic by anonymously answering a number of objective questions. Questions can be of the multiple choice type, multiple response, or true/false, although other question types may be introduced, such as gap filling, ranking, numerical responses, and visual
In this demonstration, you provide the amplitudes and frequencies of message signal and carrier, cut off frequency of LPF, choose noise on/off, demodulation phase influence, domain and scale, then it will generate the image with those effects.

Select your settings and click on "Plot". (Click the diagram for details in course notes)

Figure 2  Graphical user interface for interactive tutorial demonstration.

Figure 3  Results generated from user settings in Figure 2.
identification. These types of questions lend themselves easily to automatic marking. Due to the technical nature of the material, it was required that the assessment system be capable of handling multimedia content, such as figures, images, and mathematical expressions. The learners would be given immediate feedback after submitting an answer, and a grade at the end of the assessment. Hints, advice, and links to further reading, tutorials, and other learning resources would also be provided.

**Implementation Approaches**

There are also various methods to implement such an online assessment system, such as direct HTML authoring or using CGI scripts. Several different methods have been considered and below are some of the merits and drawbacks of the more common implementations.

The simplest implementation would be to produce each question using an HTML authoring tool, in a manner similar to producing a standard examination paper. De La Salle University [3] has adopted such a system. The simplicity of such a system comes at a price: it would be extremely difficult to manage and maintain a set of such questions, and the learner would receive no feedback at the end of the test. Indeed, as the sequence of questions and answers would be fixed, such a non-intelligent system would quickly lose the interest of most learners.

CGI scripts are often used to add intelligence to such pages, but as these scripts run on the server, this would necessitate users’ browsers constantly connecting to the server to load questions and check responses. The frequency of connecting to server would be much higher than that in the interactive demonstration system, and the ratio of network transmission time to program processing time would be high. Not only would this cause delays for the user, but it would also burden the server unnecessarily.

Web page scripting languages such as JavaScript or VBScript could be used, as these are interpreted within the users’ browsers. The major gain form local processing would be in speed and responsiveness. But as scripting code is included within the HTML document, a major drawback of such an approach is that one would have access through one’s browser to the HTML source code, and hence to the answers to the questions. Furthermore, as mentioned before in the direct HTML implementation, questions and choices are fixed after the scripts are written, leading to the problems mentioned above relating to maintenance and updating.

Most of these shortcomings have been avoided by implementing the system as a Java applet. As applets run from within the user’s browser, they are capable of checking answers and providing feedback without requiring frequent connection with the server. Furthermore, as the code included in the document is a compiled class file, the answers are thus concealed from the user. Questions are selected at random from a question database, and are displayed in HTML, with the choices also in random order.

**Question Database**

A question database was created for the module instructor to deposit questions. As mentioned above, one of the design objectives was to allow multimedia content within these questions. At present, a simple text file database structure was used, whereby special effects such as mathematical expressions and figures can be included by using a number of special characters. Using such a database, the Java applet is responsible for converting the text contents into HTML format for displaying on Web browsers associated with the applet-based graphical user interface. Examples of these are shown in Figure 4. A design decision should be made here: whether to write the converted HTML documents to the server or the client.

**File Management**

One of the intentions of applying Java applet was to reduce the frequent connections between client and server. However, writing the converted HTML documents onto the server would defeat this purpose because of the repeated communication between them when reading from the database and writing HTML files to the server. Moreover, different temporary HTML files should be taken into accounts for multiple simultaneous connections. Another problem is that this would leave the server vulnerable to malicious attack if the applet were given write-access to its file system.

Thus, this MCQ system adopts writing HTML documents onto the client, as shown in illustrated in Figure 5. Not only are all the tasks performed at the client side but also the temporary HTML files are written onto the client hard disk where the browser saves temporary files. Once the applet is downloaded, the browser at the client side needs only to connect the server once at the start of each assessment session. The applet responds by downloading the question database from the server, selects a random subset of these (section and quantity determined by the user),
and converts them one at a time into HTML. So while questions are being displayed, the client browser is reading documents from the local hard disk. Checking answers and providing feedback is also performed locally within the applet. Time delay has thus been considerably reduced, as has the danger of writing to the server.

Although the tests are taken anonymously, the result statistics are sent back to the server in order to provide the module instructor with useful feedback about the learners’ progress, such as an indication as to which questions are most frequently answered incorrectly, or which sections are most problematic.

**COLLABORATION AND INTERACTION**

Proponents of interactive technology claim that collaboration and interaction are important com-
ments of course design [4]. These ingredients can help instructors avoid the pitfalls of technology courses that rely on information acquisition and repetition of rote answers. Interaction can take place via individual email communication, or through the module bulletin board. The latter is a simple but effective web forum to allow students to post their work, ask questions, read the work of their peers, and engage in course-related dialogue. Feedback about the WBL system can also be provided here.

CONCLUSIONS

Preliminary observations concerning this system are that it met many of both the academic and technological expectations. Within a relatively short time, it was possible to provide learners on this module with a web-based tutorial and assessment system to support the existing taught elements. Although considerable work remains to be done in areas such as the question database, this attempt has demonstrated how it is possible to use simple tools to construct a fast, robust, and effective learning aid, without necessarily possessing much specialised programming skills. The system is readily adaptable and will form the basis of further web-based learning systems within the Department.

REFERENCES

BIOGRAPHIES

**Waleed Al-Nuaimy** graduated in 1995 with a degree in electronics engineering and communications from Baghdad, Iraq, joining the University of Liverpool in 1996 for a doctorate in electrical engineering. His research interests include communications and signal processing, and he is currently involved in developing signal and image processing techniques for the fusion, analysis, and interpretation of geophysical data. He joined the Department of Electrical Engineering and Electronics as a lecturer in 1999 after working for a year as research principal at Geo-Services International (UK) Ltd. Other interests include medical image processing, software radio, and the development of distance learning using Web-based teaching and learning tools.

**Jinghua Zhang** was born in Guangdong Province, People’s Republic of China in 1976. He received his BEng with a major in electronic engineering from Shenzhen University, China, in 1999 and his MSc(Eng) in intelligence engineering with distinction honors from the University of Liverpool, UK, in 2000. He developed a simulation routing software for ATM networks as his BEng final project, and this Web-based learning environment was his MSc project. He is currently pursuing a doctorate degree at the Centre for Intelligent Monitoring System (CIMS), University of Liverpool. His research topic is early identification of incipient transformer failures due to partial discharges.

**Alan Noble** is a senior experimental officer in the Department of Electrical Engineering and Electronics at the University of Liverpool, UK, working as part of the IT Support Group. He joined the department in 1981, designing 8-bit microprocessor systems. Over the years, he has moved onto distributed parallel processing systems, designing hardware and software interfaces. His current responsibilities are largely involved with Windows NT networking and Internet/Intranet development. His research interests include development of interactive teaching and learning tools.