World Wide Web Approach to Teaching Microprocessors

Francis L. Merat and Dukki Chung
Department of Electrical Engineering and Applied Physics
Case Western Reserve University
Cleveland, OH 44106

Abstract - *We have implemented World Wide Web based quizzes for an introductory microprocessor course. Our approach can be viewed as following that of an asynchronous learning network (ALN) model and has focused upon improving learning in a course taken by large number of undergraduate engineering students. Our work was tested on a subset of this class during the Fall 1996 semester with additional testing planned for the Summer and Fall 1997 semesters. We describe the initial, limited use of these electronic quizzes and make some tentative conclusions about the design of Web based quizzes.*

Introduction

Normal Computer Assisted Education (CAE) tools have traditionally been developed for the multiple answer question formation. However, much of science, math and engineering relies upon numerical problem solving which, although it can be tested in a multiple question format, is better suited to parametric problem solving. Using a parametric approach and the World Wide Web, a general problem is coded by the instructor, a computer generates a random set of initial conditions for the general problem and generates a specific quiz problem which can be delivered to the student via the World Wide Web. The student answers the problem in an open book, unlimited time format and enters his/her numerical solutions. The results are then submitted to an appropriate Web server where the program is graded (based upon the original set of initial conditions), the grade recorded in a secure database, and the grade almost immediately returned to the student.

There are many programs which can be used for multiple question problems. Also, there are many simulation programs such as Interactive Physics which can be used in an interactive manner. What is missing in computer assisted education is a homework/quiz module which can provide timely feedback about student progress to both the instructor and the student. The Electronic Laboratory Simulator [1] is one of the few programs which tries this concept, but the problems are hard coded for a particular course and cannot be really changed; also the grading module does not gracefully handle a loss of network connection. Oakley has recognized the lack of a homework/quiz module and is developing software similar to ours [2, 3]. His key concept is that of Asynchronous Learning Networks (ALN) where he utilizes a program to generate questions for the students and, once submitted and graded, students are encouraged to answer questions about solving the problem using an Internet chatroom.

Our approach can also be viewed as an ALN model and has focused upon improving learning in a large introductory microprocessor course taken by nearly 180 undergraduate engineering students per semester. Such a large course has a significant problem with timely feedback. If the students do not get any feedback until their homework is graded (at least several days for a large class), the educational value of doing homework is low. We are developing methods of generating individualized quizzes which can be given and quickly graded using the World Wide Web as a delivery vehicle thus fulfilling the educational role of more traditional homework in smaller classes.

Our quizzes are implemented in the form of HTML template documents, i.e. a form, where each question has a solution which can be represented as a mathematical relationship. Whenever a student uses a browser to take a particular quiz problem, the problem conditions are changed and each student is given a unique problem although the general structure of the problem remains the same. The student enters his/her answer to the problem on the form and electronically submits the form to the department Web server. The Web server automatically grades the submission and almost immediately returns a grade, along with a set of correct answers, to the student. Using this approach, students can quickly identify material which they do not understand and repetitively drill that particular set of problems. This computer based approach provides immediate feedback for all students in the course, regardless of class size, and information which can be used by the instructor to evaluate student learning.

Background

Much use has been made of the Internet in classes. Historically the first use of the Internet was to improve communications between the student, his/her classmates, and the instructor. The next step was to provide supplemental information such as reference material through FTP sites and Web pages. The next stage of Internet usage, which is happening now, appears to be using the Internet in an interactive manner using video conferencing, chat lines and other similar tools. We are developing a new interactive use of computers and computer networks to provide individualized, electronic quizzes with the ultimate goal of improving the learning process in higher education.

At Case Western Reserve University, all student dormitory rooms have direct optical fiber Internet connection. The campus also has over 100 modems in a modem pool supporting PPP connection to the Internet; and various campus computer labs support clusters of Internet connected personal computers and workstations. As a result, CWRU students have access to the Internet at all times.

In our EEAP 282 (Introduction to Microprocessors) class, we have been very aggressive in using the Internet and the World Wide Web for educational purposes. In the past we have placed over 400 pages of reference materials and notes in HTML format on a class Web Site, we have used a bulletin board to stimulate class related discussion, and we have extensively used e-mail to answer questions and dialogue with students. Because this is a large class (typically 180 students) we have identified problems peculiar to using the World Wide Web with a large number of students. Specifically, the World Wide Web enables individual communications but, in large classes, the volume of this information can be overwhelming. During the Fall 1996 semester, the EEAP 282 bulletin board was among the most used on campus. The instructors and teaching assistants received and responded to over 3500 student e-mails related to this course. This volume of information is quite overwhelming and does not lend itself to conventional use of the Web.

Our development of Web based quizzes was based upon a realization that we could no longer assign and grade homework in any conventional sense. Our goal using on-line quizzes was to replace the homework formerly assigned in the class and develop a more timely method of grade feedback. To achieve this goal we have developed some prototype on-line quizzes [4]. Students can log in to a Web server using their campus network IDs and passwords and select one of several topical areas to be quizzed about. Once a topical area is selected an individualized quiz consisting of approximately 10 problems is generated, the Web server generates a record of the problems generated, and a Web form containing the questions is generated and transmitted to the student. The resulting quiz is open book and untimed. When the student fills out the form with his/her answers, he/she submits the form to the Web server which immediately grades the answers, and almost immediately transmits a grade plus a comparison of the student's answers and the correct answers back to the student. Our questions are currently unambiguous in terms of the answer, i.e. the grading is binary and does not evaluate "close" answers. The educational goal is to enable quick feedback to the students about their abilities and to identify where additional study and help is needed.

Implementation

Although the software could be developed for almost any machine we have used our department Web server (an HP 9000 C-Class Web Server) coding the software in a combination of C++ and HTML. An immediate problem we encountered was authenticating who was taking a quiz. Our initial tests of the Web quiz concept immediately led to students taking quizzes for other students as the quizzes were typically taken in the student's dorm itory room. We have since implemented a more secure "Kerboros"-based security module which uses the student's campus network ID and password (far more unlikely to be shared with other students) and provide a higher level of student authentication. Sharing of university computer accounts is prohibited by Case Western Reserve University's Computing Code of Ethics. Also, students are very reluctant to share computer accounts and especially passwords. If the Kerboros module by itself does not prevent most authentication abuse we can always give the quizzes in a supervised computer lab however we do not anticipate such a drastic step.

The question generation module utilizes a combination of text and random number generation. For example, consider the following simple question:

> Indicate the number of bytes of memory allocated by the following assembler directive.

DS.L 5		bytes

The directive DS.L may be considered as the rule for generating the question. The number 5 is the argument for the instruction and is the initial condition. The answer in terms of number of bytes occupied in computer memory by the instruction is computed by the program, and is stored as the answer for the problem. The student is presented the problem as shown above in the form of a Web form and the student enters the number of bytes. Upon pressing the submit button, the answer is sent to the server where it is compared with the correct answer, a grade determined and recorded, and the grade returned to the student. Although rather trivial the above example is indicative of the types of questions that can be generated and answered using a Web compatible format. Much more complex questions are possible, i.e.,

> Given that PC=$39A5, SP=$432C, and the stack is initially clear, show the stack contents after the instruction at $3BA5 executes.

```
ORG       $39A5
JSR.S     $3AA5
ORG       $3AA5
ADD       #-2,SP
MOVE.W    #$FBA3, (A7)+
JSR.S     $3BA5
ORG       $3BA5
MOVE.W    SP,-(A7)
```

Address	Contents
$4320	
$4322	
$4324	
$4326	
$4328	
$432A	
$432C	
etc.	

Figure 1. Sample Question

This is not a multiple answer format, but this type of question can be formulated and graded by a computer just as the previous example was.

In practice the HTML document for each topical area is generated by a C++ program when the corresponding HTML link is activated. Some numeric quantities in each question are changed by using a random number generator, for example, "DS.L 5" might be changed to "DS.L 8" for another student accessing the quiz. In this way, each student can get his/her own customized quiz. The change in the problem will also be reflected in the corresponding grading module. When the student clicks the submit button after finishing the quiz, the form is transmitted to the grading module. The grading module grades the quiz, stores the grade in a file, and immediately returns the result to the student. Unfortunately the present format requires "hard coding" of each problem resulting in a lot of work for the instructor (and teaching assistants) and a limited number of quiz problems actually being generated. A more modular method of entering problems is under development.

Sample Interaction with Electronic Quiz

The following examples illustrate some of the steps involved in working through a typical quiz. Students can use any modern Web browser to take and submit their quizzes. First, the student must access the class Web page and choose a specific quiz from an index of topics.

After choosing one of the possible quizzes the student logs in with his/her campus network ID and password. After logging in the student is presented with a quiz similar to that shown in Figure 2.

The student must now solve the problems and type his/her answers into the blanks provided as shown in Figure 2. After answering all the questions the student clicks on the "Grade HW" button to submit the homework for grading. The grading module on the server then grades the submitted homework and almost immediately returns the results to the students along with a set of correct answers. The results are also recorded in a central database so that the instructor can retrieve and examine the results.

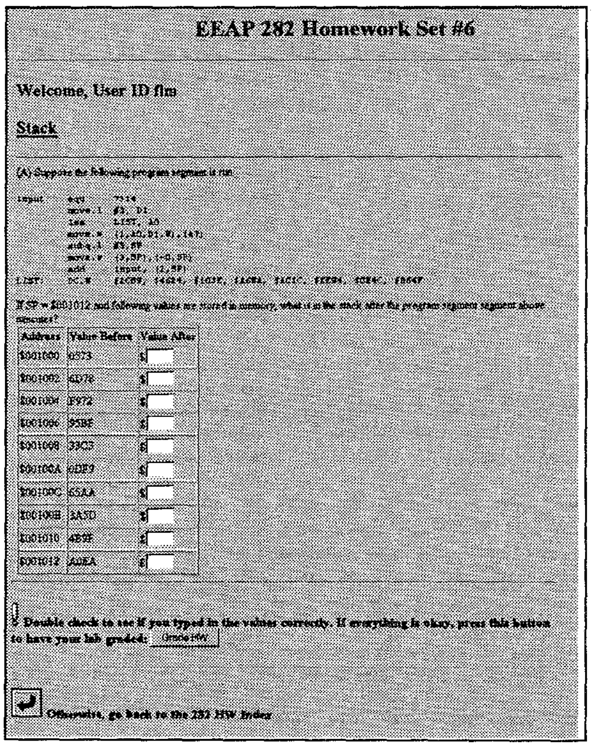

Figure 2. Electronic Quiz

Issues Regarding Electronic Quiz

We have tested this software on a voluntary (and very limited) basis during the Fall 1996 semester and have gotten some useful student feedback about our initial prototype. The students feel that the educational goals seem reasonable and generally liked the quiz format and problems. They especially liked it as a study aid for exams. However, much more work is needed to make the on-line quiz concept technically and educationally viable. Technically, we need to:

1. keep the individualized grading of the problems confidential from other students using appropriate authentication. A Kerboros module is being tested.
2. break the quizzes down into individual problems which are identified by topical area and difficulty. Our initial quizzes were in a standard quiz format containing ten problems. A major problem was that the students felt that the problems were too interdependent and requested individual, standalone problems.
3. develop a standardized method for the instructor to enter new problems or change existing problems. Far too much time is spent entering and debugging a quiz problem for this method to be viable for anyone except a stalwart programmer.

Discussion and Conclusions

We have used the World Wide Web as a vehicle to produce individualized quizzes to provide feedback to both the instructor and student about student comprehension of numerical problem solving skills.

Our approach has been to identify and test using the Web as an instructional multiplier for large classes. To this end we have identified two goals which can be effectively delivered over the Web: (1) individualized quizzes to be accompanied by supplemental, tutorial materials; and (2) on-line individualized labs. The first has been described in this paper. The second is currently under development for our introductory electronics class where we are using the same approach described here to generate individualized student labs. Here the basic circuit topology is the same for every lab group; however, every lab circuit uses different circuit values and is slightly different. Students record their data on a Web form, submit it, and a program checks the data for consistency with the given circuit elements. The grading module includes a simple circuit simulator which evaluates how close the submitted data is to the correct value. Students work is graded only as acceptable (with a given tolerance of the correct answers) or unacceptable (too far from the correct answers). This work is still in the development stages and has not yet been tested in a real class setting.

The goal of this work is educational. The authors teach EEAP 282, a very large introductory programming class taken by slightly under 200 students, which is also the test course for this work. The major difficulty involved in this course, which is very problem oriented, is that students can watch the instructor explain concepts or solve sample problems and think that they understand concepts, but when they attempt to solve different problems on their own they get stuck part way through the problem. Traditional homework assignments are not adequate as the students do not receive feedback in a timely manner. A few problems can be assigned; however, grading 200 student assignments may take several days or more. With the proposed software tools utilizing the Web the emphasis is still upon the instructor to prepare the quiz question(s); however, the grading is electronic and feedback is delivered to the student (and the instructor) in a much more timely manner.

We are in the process of researching the type (format) of questions that can be presented to the students using the Web format, developing a standard model or set of models for representing such questions, and developing an editor for the instructor to quickly enter and verify the correct presentation of a student problem, rather than spending most of his/her time programming a simple question.

Acknowledgements

The authors wish to acknowledge the contributions of Dennis Lee who coded the original software in HTML and C++, Kurtis Swinehart who is recoding the software in PERL, Wes Brown who developed the Kerboros module, and Robert Delvalle who is testing the quizzes in his summer EEAP 282 class.

References

1) Electronic Laboratory Simulator, *Falcon Software*, Wentworth, NH (ISBN 1-886959-02-1).

2) Oakley, B. II, "CircuitTutor", in *101 Success Stories of Information Technology in Higher Education: The Joe Wyatt Challenge*, J. V. Boettcher, Ed., McGraw-Hill, 1993.

3) Oakley, B. II, 'The Virtual Classroom: At the Cutting Edge of Higher Education", in *Frontiers in Education Conference*, Salt Lake City, Utah, Nov 1996.

4) Sample quizzes are at http://dumbo.cwru.edu/dept/courses/282/newlab/282index.html. Use demo as an account name to get access to the quizzes.