

# DISTRIBUTED LEARNING VIA THE WORLD WIDE WEB THROUGH INTERACTIVE MODULES

*John W. Lockwood, PhD*

Department of Electrical and Computer Engineering  
University of Illinois  
Urbana, IL 61801  
lockwood@ipoint.vlsi.uiuc.edu

## ABSTRACT

This year, new distributed learning tools were developed and integrated into an electrical and computer engineering course at the University of Illinois. This paper discusses the use of the World Wide Web (WWW) for the teaching of ECE291 and describes the development and implementation of new interactive software modules that automatically grade on-line homeworks, record student scores, graphically display point distributions, and manage group projects.

## 1. INTRODUCTION

The ECE291 course at the University of Illinois instills fundamental concepts in computer engineering by teaching assembly-language programming for the Pentium microprocessor. This class helps students conceptually bridge the gap between digital circuit design and programming with high-level languages. Students learn how data structures are stored in memory and how algorithms are implemented at the machine level [1].

Starting in the Fall of 1996, new on-line content and interactive software modules were developed as teaching aids for ECE291. Using standard World Wide Web (WWW) browser, students can now access the class from any machine on the Internet. Section 2 begins by describing the course materials and lecture notes now available on-line at the ECE291 home page [2]. Section 3 describes the interactive software modules that have been developed for this class. These tools automatically create the student database, grade on-line homeworks, record scores, graphically display point distributions, and manage group projects. Section 4 details how these interactive tools were implemented using Common Gateway Interface (CGI) programs and a Structured Query Language (SQL) database. The paper concludes by showing how the minimal computational and network resources required to run this software have increased the quality of the class by providing immediate feedback to the students and giving the teaching staff more time to interact with the students.

## 2. NON-INTERACTIVE MATERIALS

Standard Hypertext Markup Language (HTML) documents are effective for distributing non-interactive course material to students. Within the last year, all of the ECE291 lecture notes and course materials have been made available to our students and the to the on-line community as a whole. The lecture notes cover all ECE291 topics including:

- Computer history
- Assembly-language programming of the 80x86

- Programming with the stack and recursion
- Handling interrupt-driven input and output
- Interfacing the CPU to hardware devices
- Floating point math Unit and FPU calculations
- SIMD using MMX instructions

In total, the lecture notes contain 25 sets of HTML documents with dozens of figures and diagrams, representing 40 hours of lecture material.

Previously, students purchased copies of handwritten transparencies from copy shops on campus. Now, students obtain class notes directly from the ECE291 web server. On evaluation forms, students noted the following advantages of the on-line lecture notes:

- On-line notes are readily available.
- On-line notes can be read from anywhere.
- Typeset text and diagrams are legible.
- On-line notes are free.
- Hotlinks speed finding information.
- On-line notes are always up-to-date.

## 3. INTERACTIVE ON-LINE MODULES

While web documents are well suited for the presentation of lecture material, they cannot be used alone to present material that changes dynamically. Interactive modules have been developed that implement a gradebook, electronically grade on-line homeworks, graphically display score distributions, and track membership in class projects. The functionality of these modules is described in the following sections.

### 3.1. Student database

At the beginning of the semester, each student uses a web browser to enter information about himself or herself into the on-line database. Each student provides his or her name, major, academic year, and selects a unique, 5-character, alpha-numeric public ID. Collectively, the information entered by the class becomes the student database and creates the on-line gradebook.

### 3.2. On-line gradebook

The gradebook tracks the scores for all homeworks, machine problems, and exams. A public version of the gradebook is available on-line at all times and can be accessed directly from the ECE291 homepage. Accessing the page implicitly queries the gradebook database to produce an up-to-the-second summary of all scores sorted by IDs. Students can gauge their performance relative to other members of the

class, as all scores are visible. The use of the public ID preserves student anonymity.

### 3.3. On-line homework

The on-line homework module allows students to electronically submit homework from a web browser. The answers to the homeworks are graded automatically. The grading module awards points for correct answers and offers context-sensitive help for answers that are incorrect. When a homework is submitted, the student's highest score is immediately posted to the gradebook. Students are allowed to resubmit homework and earn additional points.

All ECE291 homeworks submitted during the last two semesters have been graded electronically. The online homework has been very popular with the students because it provides instantaneous feedback on their answers. A survey of the students' attitudes towards homework revealed that an overwhelming 92% of the students strongly favored the on-line homeworks, 4% were neutral, and 4% would have preferred to revert back to paper assignments. This homework module has also been very popular with the teaching assistants as it offloaded the tedious task of homework grading to a machine. Hours that teaching assistants previously used for grading homework are now used for helping students in the programming laboratory.

### 3.4. Manual score entry

Exams and some assignments for ECE291 are still graded by hand. The administration module allows these scores to be manually entered into the database. Using a web browser and a password, the instructor and teaching assistants can access the administration menu. From this menu, the *grade* option alphabetically lists student names and provides input boxes for entering new scores. When new scores are entered, they are immediately posted to the gradebook. For security, all administrative actions are also recorded in a log file.

### 3.5. Score distribution plots

Undergraduate students feel compelled to know how they are performing as compared to rest of the class. Using the on-line gradebook database, the distribution module creates a distribution plot of the scores. The output of this module is a bar chart that shows the frequency of student scores. Students can graphically see how their total score or individual scores compare to the rest of the class.

### 3.6. On-line final project signup

The final machine project for ECE291 is a team project. Students are given the freedom to create their own project or join an existing team. The on-line project module tracks membership in these teams as they are created. Students wishing to create a project can post a new entry to the list. Students wishing to join a project can browse the list. Once the team membership has been established, the project module is also used to award points to individual members.

## 4. IMPLEMENTATION MODEL

Clients (students) use standard web browsers (such as Netscape, Internet Explorer, or NCSA's Mosaic) to query the gradebook database and submit on-line homework. Common Gateway Interface (CGI) programs written in C++ run on the server and produce output that is interpreted as simple HTML by the client. Because database queries and grading are done by the server, no additional software needs to be installed on the client machines. Such

an approach avoids the complexities of platform incompatibilities and version control that could hinder students from using the tools in the first place.

The ECE291 CGI programs are responsible for grading the homework, plotting grade distributions, editing project data, and updating scores. CGI programs display results as standard HTML tags and read input as fields from a form. Each time that a CGI program is invoked, the web server forks separate process to handle the request. Therefore, at any given time, it is possible that multiple CGI programs are running simultaneously. Structured Query Language (SQL) queries are transferred via sockets between CGI programs and the database engine. The SQL engine serializes transactions and manages the common databases.

Directory-level passwords (as implemented by the web server) protect private information and prevent unauthorized modifications to the database. No individuals other than the instructor can view private information or update scores. For public networks, the Secure Socket Layer (SSL) can be used to encrypt all data before it is transmitted over insecure media.

## 5. COMPUTATIONAL RESOURCES

All of the components for the ECE291 on-line modules are based on non-proprietary tools that have no cost to use for academic purposes. An on-line server need only consist of the following components:

- Host computer
- Web server
- SQL database engine
- ECE291 CGI programs

Our existing configuration is implemented as

- 5x86 PC running Linux
- NCSA httpd web server
- mSQL database engine
- ECE291 CGI programs

Even during peak usage for our class of 85 students, the average CPU utilization of the server remains under 5%. If desired, the implementation can be compiled to run under nearly any UNIX or WindowsNT platform.

## 6. SUMMARY

Experiences with the World Wide Web both for distributing learning and interacting with students has proven to be very favorable. The interactive tools described in this paper are currently used for the ECE291 course at the University of Illinois. Students strongly approve (and almost enjoy) the on-line homeworks and noted that they appreciate the immediate feedback provided by the automatic grading module. The software tools described in this paper were built as CGI programs written in C++ atop non-proprietary web server and SQL database engine. These tools are portable and could have utility in other teaching environments.

## REFERENCES

- [1] Michael C. Loui, "The Case for Assembly Language Programming," *IEEE Transactions on Education*, vol. 31, no. 3, pp. 160-164, Aug. 1988.
- [2] The ECE291 Homepage at the University of Illinois, <http://www.ece.uiuc.edu/~ece291/>.