Java Programming for Engineers: Developing Courseware for a Computer-Enhanced Curriculum

Julian A. Bragg, Clinton D. Knight, and Stephen P. DeWeerth
Georgia Institute of Technology School of Electrical and Computer Engineering,
Atlanta, GA 30332-0250

The School of Electrical and Computer Engineering at Georgia Tech has embarked on a Computer Enhanced Education (CEE) initiative to augment the core ECE curriculum (courses in signal processing, circuits, digital systems, microelectronics, and electromagnetics). One of the goals of this initiative is the development of interactive, graphically based software that helps students gain intuition about the physical basis of many of the core ECE topics, an intuition that is often obscured when the topics are approached from a purely mathematical standpoint. The Java programming language provides an ideal platform for the development and for the dissemination, both in and out of the classroom, of this courseware.

In order to both stimulate the development of the CEE courseware and to introduce students to the powerful capabilities of modern programming languages, we have developed a course titled *Java Programming for Engineers*. This course introduces students to advanced programming concepts by exploring their use in the Java language. The instruction is leveraged in such a way that the students produce, as a final project, an engaging, interactive demonstration of a basic ECE concept; many of these demonstrations have already found their way into the classroom in other courses. The result is a course with dual, complementary goals: teaching Java-based programming concepts and developing computer-enhanced educational courseware.

I. Motivation

Engineering topics, especially those in Electrical and Computer Engineering, are often taught by first presenting the underlying equations and then, by exploring those equations, revealing the topic's fundamental principles. This approach, while common, has the unfortunate effect of postponing intuitive understanding of the topic until well into the process, when the student has mastered the mathematics to the point that the "big picture" can be seen. In fact, many students learn the equations quite well, but never reach the point of truly understanding the principles illustrated by those equations.

A major goal of the CEE initiative is the development of courseware, and, in particular, software that allows students to gain an initial intuition about a core ECE topic in conjunction with the presentation of the equations. Thus, as they struggle with the often complex mathematics underlying that topic, they have the advantage of understanding how those mathematics fit into the system as a whole.

The one-on-one nature of educational software gives it great potential as a tool to help students gain intuition about a topic. After presenting a typical engineering scenario in a simple, graphical manner, a program could allow a student to change parameters and see immediately how their changes affect the system. Far more than a static picture or an animation, educational software offers the possibility of a truly interactive learning environment.

II. Course Content

Java Programming for Engineers has two primary student-produced components: (1) a series of programming assignments that illustrate the capabilities of the Java language, and (2) a design project that applies Java to computer-enhanced education. We focus the course material on Java capabilities—object-orientation, graphical user interfaces, event-driven programming, multithreading, and networking— that have particular utility for educational applications. Although these characteristics add extensive functional potential to the language, they also represent a new programming paradigm that makes the Java learning curve particularly steep for students who are familiar only with traditional functional programming languages (e.g. C, Pascal, and Fortran).

In order to reduce the Java learning curve, we structured the homework assignments in such a way that students are using traditional functional, programming styles (e.g. infinite loops) at first, and gradually moving towards producing a fully graphical, event-driven program. The first homework assignment requires the students to modify a traditional functionally programmed loop that drives an instructor-provided applet. We provide classes that take care of the graphical user interface, event handling, and other complex aspects of the program, allowing students to practice Java fundamentals and produce their first applet without having to understand more advanced Java topics. With each successive homework assignment, students add functionality to their applets, while removing some of the instructor-provided “crutches”. The final assignment, consisting entirely of their own code, is an applet making use of object-oriented design, event-driven programming, drawing, animation, multithreading, and networking—all within a graphical user interface.

In addition to the homework assignments, students (in teams of two) complete a software design project in which they choose a particularly “confusing” concept from the core ECE courses and develop a Java applet that demonstrates the concept in an intuitive manner. Each team is required to consult with a faculty mentor who teaches the core course in their project area. This mentor advises the team on the applicability of their project to this course and often continues to work with them after the course ends to further develop the software. Due in part to this mentoring structure, many of these projects have already found their way into core ECE course materials, either as in-class presentations, or as one-on-one demonstrations incorporated into class web pages.

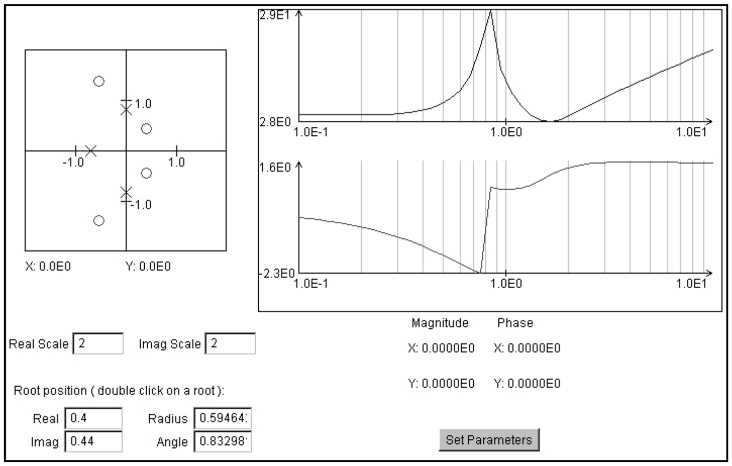
III. Project Incorporation into Core Curriculum

Java-based CEE courseware has been included in all of the five core disciplines in the Georgia Tech ECE curriculum. For example, electromagnetics applets that demonstrate the actions of charged particles in a plane or standing waves in a transmission line are helping students gain an initial intuitive feel for these often intimidating topics without being swamped by the mathematics. The digital-signal-processing course is using applets illustrating aliasing and the principles of Fourier analysis, while other courses are using applets that examine mass-spring systems and lens principles. Many of these applets can be seen at the CEE web site (http://www.ece.gatech.edu/academic/computer_education/demo_java.html).

In addition to generating a constant flow of new courseware projects, previous offerings of *Java Programming for Engineers* have created a large number of students familiar not only with the Java language, but also with using Java to generate interactive educational software. Many of these students have gone on to refine class projects and develop completely new ones, either as research projects or while being funded by the ECE initiative.

To stimulate additional courseware development, a group of students funded by the CEE program has been developing Java components for incorporation into future projects. For example, they are developing a generalized plotting component that will automatically display both continuous functions and discrete data points. Future courseware developers will be able to use this package to perform graphing functions, rather than having to recreate a charting component from scratch.

The CEE student group has also been investigating the use of Java Beans technology to allow rapid development of courseware. As a proof of concept, they have developed a set of Beans that can be used to present s-plane concepts. By combining these Beans in different patterns, applets have been produced that allow the relationships between s-plane pole-zero plots, transfer functions, and time-domain responses to be investigated in an engaging, interactive environment. A screenshot from one of these applets is presented in Figure 1. Future Beans will allow students to lay out simple RLC circuits, and see how changes to circuit values cause changes in the system response.



X: 0.0E0 Y: 0.0E0

Magnitude Phase
X: 0.0000E0 X: 0.0000E0
Y: 0.0000E0 Y: 0.0000E0

Real Scale 2 Imag Scale 2

Root position (double click on a root):

Real 0.4 Radius 0.59464:
Imag 0.44 Angle 0.83298'

Figure 1 - The S Plane applet

IV. Conclusions

Java Programming for Engineers has proven to be an effective tool for teaching programming concepts embodied in the Java language, while developing useful tools for teaching the core undergraduate ECE curriculum. The course has been particularly well received by students (as based upon their end-of-quarter course evaluations) and has already led to the production of a number of courseware applets, many of which have found their way into the core curriculum. In addition, the class has created a group of ECE students with extensive Java programming skills and experience in using those skills to create interactive learning software (and other Java software). This group of students will continue to produce more educational applets, both by refining existing class projects and by creating new packages requested by course instructors. Our expectation is that other faculty, once they have seen the benefits of incorporating this courseware into their curriculum, will become involved with the course as mentors, bringing new project ideas and sustaining the momentum of the CEE initiative.

In the fall of 1999, Georgia Tech will be switching from a quarter-based to a semester-based calendar. Initially, this course was not expected to be included in the semester curriculum. The positive response to the course and high student demand for future offerings has convinced us to create a semester version of the course. The new course will be a greatly enhanced version of the present offering because of the extended term length (from 10 weeks to 15 weeks) and because Java fundamentals will be taught in the introductory Computer Science course that our students all take. We will expand the course to include Java extensions that are of particular applicability to engineers (e.g. Remote Method Invocation, Java Native Interface, and JINI), facilitating homework assignments and projects in areas such as hardware interfacing and embedded programming. The resulting course should strengthen the programming foundation learned by the students, while expanding the CEE efforts to areas such as remote laboratory delivery.

JULIAN A. BRAGG
Julian A. Bragg is a MD/PhD student currently doing doctoral work in bioengineering at the Georgia Institute of Technology. His research interests include biological motor control and the use of analog VLSI to model biological systems. Julian was the TA for two sessions of Java Programming for Engineers.

CLINTON D. KNIGHT
Clinton D. Knight is a graduate of the Georgia Institute of Technology Electrical and Computer Engineering PhD program. His thesis work involved the use of embedded systems and the Java programming language to allow EE laboratory assignments to be completed over the WWW. He was the lecturer for the first offering of Java Programming for Engineers.

STEPHEN P. DEWEERTH
Stephen P. DeWeerth is an Associate Professor of Electrical and Computer Engineering at the Georgia Institute of Technology. His research interests include the use of neuromorphic analog VLSI to model biological sensorimotor systems. He has been the instructor of Java Programming for Engineers since its inception.