Web-Based Enrichment Courseware for Introductory Engineering Students

SALLY WOOD
Electrical Engineering Department
Santa Clara University
Santa Clara, California USA

RONALD DANIELSON
Computer Engineering Department
Santa Clara University
Santa Clara, California USA

ABSTRACT

Available computing and communication technologies, coupled with platform–independent delivery systems, offer great potential for development of new types of courseware in support of engineering education. This paper presents both an analysis of general issues related to development of such modules and examples from a collection of such courseware being developed to help introductory students learn fundamental concepts. This experience demonstrates the importance of development tools and improved adaptability for courseware development.

KEYWORDS

Java, interactive courseware, undergraduate engineering education, dynamic visualization

INTRODUCTION

Web-based interactive software coupled with high performance browser systems offer great potential for delivering course material that allows students to move easily among concepts in response to their level of understanding and their interests. Multiple media can provide a variety of visualization aids to better match the different cognitive styles of the students and provide a much richer learning experience than a "one size fits all" text book. The interactive component allows students to control their exploration of new material in many more ways than the predominantly passive reading of text and reference books. Evidence suggests this hypermedia approach is particularly relevant for beginning students [1]. Exactly how this potential will be realized is of interest to educators from kindergarten to the university level.

Rapid technological advances over the past ten years have dramatically altered the type of courseware that can be created with a reasonable investment in development time. Prior to windows–based operating systems, the major effort of any educational software development project was the creation of an effective user interface, and this overhead inhibited the creative development of courseware. Early authoring tools for relatively low resolution, limited color displays tended to focus on providing control of trajectories through hyperlinked text. This type of presentation was not substantially different from text books. As standard graphically-oriented user interfaces have evolved, courseware development efforts have been able to focus more on pedagogical content rather than details of the user interface.

In many areas of study a dynamic presentation of the material can provide opportunities for new insights in a manner that is not possible with static displays common to text books. For example, the transition from transient to steady state response of mechanical and electrical systems is not intuitively obvious from mathematical representations. However, it is much more clear when the time response of differential equation solutions is observed or physical models are simulated [2]. Similarly, the behavior of algorithms is more easily explained if a dynamic model of a sorting algorithm or a finite state machine, for example, can be observed.

Despite the advantages of this type of courseware, effectiveness is still limited by platform dependence problems. Maintaining courseware operability under the variations of configurations at educational institutions by simultaneously supporting multiple versions increases the probability of errors and is beyond the resources of most instructors/courseware developers.

The recent development of Java and widespread use of the Web allows the potential for development of courseware that can be used on any platform under any operating system. Although this potential is not fully realized at present, great progress has been made. Instructors can develop modules of their own and combine these with modules generated elsewhere to provide a rich source of supplemental material that adds new capabilities and goes beyond simply replacing older media and styles.

This next stage in courseware development raises significant other issues that must be addressed. How can generic modules be created that can be easily combined with others? What are optimal templates for effective integration of various media (text, video, audio) under interactive control? What kinds of development tools are needed to quickly and effectively create modules? The issue of courseware development tools is significant even though the computational and display resources to make custom interactive graphic displays are readily available. Much more precious is the time needed by a content developer to address a broad spectrum of presentation styles in addition to the issues of courseware content.

APPLICATION AREAS AND APPROACHES

Our overall approach follows that used in previous development of two Motif-based tutorial collections for undergraduate engineering courses [2]. Graphics, animation, or other non-textual materials help explain the topics and place them in context. The student can modify relevant parameters and explore the effect of different combinations on behavior. This presentation is supported by short segments of text providing brief overviews of the topics being explicated, with links providing access to more in–depth explanatory materials where they are available.

Figure 1, for example, shows a Java applet which allows the user to set physical parameters for a pendulum system, modeled by second order differential equations. The student can observe both the mathematical solution fundamentals and a simulation of the pendulum’s motion. Similar simulations can show the response of a spring or a circuit with different physical parameters but the same equation coefficients. The effect of physical changes of gravity and length can be observed in simulation behavior and mathematical solution.

We are initially targeting three courses for use of our materials. One is the introductory Java programming class for computer engineering majors, where concepts of control flow, parameter linkage, object inheritance, and object design have always been hurdles to student success. A second course is an introduction to basic concepts of electrical engineering, such as power, energy, voltage and current, sinusoidal signals, time constants, feedback, and transducers [3]. This course paves the way for subsequent circuits courses by providing a direct tie between electrical phenomena and personally observable experience. The third course covers fundamental concepts of digital technologies. It is aimed at non–engineering majors and satisfies a general university core curriculum requirement related to technology. Topics that are likely to cause students difficulty and are thus targets for animation or simulation courseware include understanding the operation of a pn junction, instruction and data flow in pipelined and superscalar processors, cache operation, and network protocols and routing. Greater detail about the materials being developed for these courses is available in [4].

We are particularly interested in exploiting both discrete (text, graphics, animation) and continuous (audio and video) media in helping students to understand the material. For example, audio output is used to demonstrate many of the concepts of information representation and frequency selective filtering in the introductory electrical engineering course, and we want to use the same medium in the relevant courseware modules. As another example, video clips of the various steps in the wafer fabrication process would greatly enliven the subject for students in the introduction to digital technologies course, both during class presentations and for enrichment. This would replace the video tape on reserve in the university's media lab.

USES AND COMPONENT DESIGN

The predominant uses for our materials is to augment traditional classroom teaching, and for supplemental enrichment outside the classroom. Web–based courseware will both replace library reserves and offer more extensive resources.

Some of our materials are used in scheduled labs, where students interact with each other while using the courseware. In this setting the virtual elements provided by the modules simulate the physical elements usually manipulated in a lab. We also use the examples to enhance classroom presentations, and during office hours to help explain concepts with which a student is having difficulties, replacing the “visual aids” teachers have always dragged into classrooms. Since the courseware modules are available on a 24 hour basis via the Web, students use them individually or in groups when studying outside of class. In these situations they function like dynamic lecture handouts.

Using courseware for multiple purposes demands that the components be modularized, and places constraints on the design of the modules. Scope and dependencies of each module must be defined before development so that several instructors, employing a variety of topic orderings, may be accommodated. It should be possible for instructors to quickly organize modules into sequences that are relevant for their particular section. Using Java applets as the delivery vehicle helps achieve this balance between interdependence for context and independence for individual course structure.

Module independence allows students to move through a collection of courseware as their personal needs dictate. For example, a student who understands if–else statements but not switch statements can go directly to relevant material. This makes the modules more valuable as supplemental study materials.

Finally, while we don't intend to create full-blown tutorial programs for all of the topics for our three target courses, the topics covered by the courseware are relevant to many other computer or electrical engineering courses, and faculty in those courses may wish to create such tutorials. This means it should be possible to embed the modules in a larger structure that provides short descriptions of the relevant concepts and opportunities for learning assessment. Again, module independence and use of Java make such reuse of components easier.

LONG TERM OBJECTIVES

We are defining some Java tools that will make it easier to develop new courseware, and expanding the library of courseware modules. As a longer term issue, we are interested in using student performance in previous courseware modules and information about student learning styles to adapt the materials to more closely fit student interests.

TOOLS

All of our existing courseware has been written in Java, making use of Java Beans to simplify certain aspects of the implementation. Developing courseware in this way is time–consuming and error–prone. One of the longer term goals of our work is to identify a set of common activities, interactions, display mechanisms, and data capture operations and then to implement tools that will simplify developing new courseware. We need this for our own use, but we also want to make developing Web–based enrichment systems easier for other faculty on campus.

One such tool might help in synchronizing presentation of the same concept in several different media. For example, in the pendulum system in Figure 1, the animation of the pendulum is synchronized with the display of the appropriate sinusoidal curves, and it's easy to imagine an instructor wanting to add audible cues for additional reinforcement or connection to a similar simulator for a second order circuit. Another example might be a Java for loop, where we want to synchronize the relevant line of a program fragment with the display of an animated flowchart for the concept. Java threads allow this, but thread programming is not for the faint of heart.

ADAPTABILITY

Students vary in enthusiasm and persistence in seeking supplemental information resources, and the poorer students (those most in need of supplemental enrichment) are the least effective at ferreting out the necessary information. Our system will provide structure and guidance to students while preserving much of the Web's traditional freedom of movement through the material.

Computer–assisted learning systems typically maintain a model of the student, containing information about topics covered and mistakes made. They also model the content and conceptual relationships in the courseware and employ a specific instructional strategy, either explicitly or implicitly in a hardwired structure. We believe the model must also include information about the student's learning styles in order to most effectively present material [5]. Moreover, although there is some indication that individual media preferences don't necessarily correlate with improved learning [6], we believe the student's media preferences should also factor into adapting the content and structure of information presented.

Obtaining enough context to develop a meaningful student model is difficult to do in the connectionless mode of access common to the Web. Gathering information on learning styles, media preferences, and media effectiveness may be even more difficult. However, in a restricted, registered class situation it is feasible, although it is difficult to track enough media choices to have confidence in the preferences of individual students.

Attempts by others to incorporate learning styles into adaptive web-based tutorial software [7] have explicitly collected that information by having students complete evaluation questionnaires. We plan to do the same, and to gather broader information on media preferences by tracking their accesses when surfing the Web [8]. We hope to find correlations that will be applicable in a broader context.

CONCLUSIONS

Current technology offers great potential for the development of new types of courseware which can make instruction more efficient and offer students a variety of options in the learning process. High speed computation and communication coupled with high resolution color display provide the basis for topic–specific animations designed to communicate concepts and dynamic relationships. The growing use of Java provides platform independence so that courseware can be widely distributed without putting inappropriate software support burdens on the developers. Given these resources, creation and deployment of courseware and courseware development tools, will lead to new options for engineering education. Based on experience from previous courseware used across all levels of undergraduate engineering education, new modules have been defined and are under development which will take advantage of platform independence and Web–based use and distribution. The effective use of these new capabilities and the definition of efficient tools for creating courseware designed for specific subject areas will be important issues for future research and development.

ACKNOWLEDGMENTS

We are grateful to Hewlett–Packard and 3–Com for donations of computing and networking hardware and software to equip our lab.

REFERENCES

[1] L. J. Najjar, Multimedia Information and Learning, *Journal of Educational Multimedia and Hypermedia*, 5(2), 1996, pp. 129–150.

[2] S. Wood, A New Approach to Interactive Tutorial Software for Engineering Education, *IEEE Transactions on Education*, 39(3), 1996, pp. 399–408.

[3] S. Wood, A Concept Oriented Freshman Introductory Course Utilizing Multimedia Presentations and Group Laboratory Experience, *Proceedings of the 1998 Frontiers in Education Conference*, 1998, IEEE Press, New York, to appear.

[4] R. Danielson and S. Wood, Stimulating Introductory Engineering Courses with Java, *Proceedings of the 1998 Frontiers in Education Conference*, 1998, IEEE Press, New York, to appear.

[5] M. J. Jacobson, C. Maouri, P. Mishra, and C. Kolar, Learning with Hypertext Learning Environments: Theory, Design, and Research, *Journal of Educational Multimedia and Hypermedia*, 5(3/4), 1996, pp. 239–281.

[6] S. Palmiter and J. Elkerton, An Evaluation of Animated Demonstrations for Learning Computer–based Tasks, *Proceedings of CHI '91*, 1991, Association for Computing Machinery, New York, pp. 257–263.

[7] C. A. Carver., Jr., R. A. Howard, and E. Lavelle, Enhancing Student Learning by Incorporating Learning Styles into Adaptive Hypermedia, *Proceedings of ED–MEDIA 96*, 1996, Association for the Advancement of Computing in Education, Charlottesville, Virginia, pp. 118–123.

[8] P. P. Maglio and R. Barrett, How to Build Modeling Agents to Support Web Searchers, *Proceedings of the Sixth International Conference on User Modeling*, 1997, User Modeling, Inc., West Newton, Massachusetts, pp. 5–16.

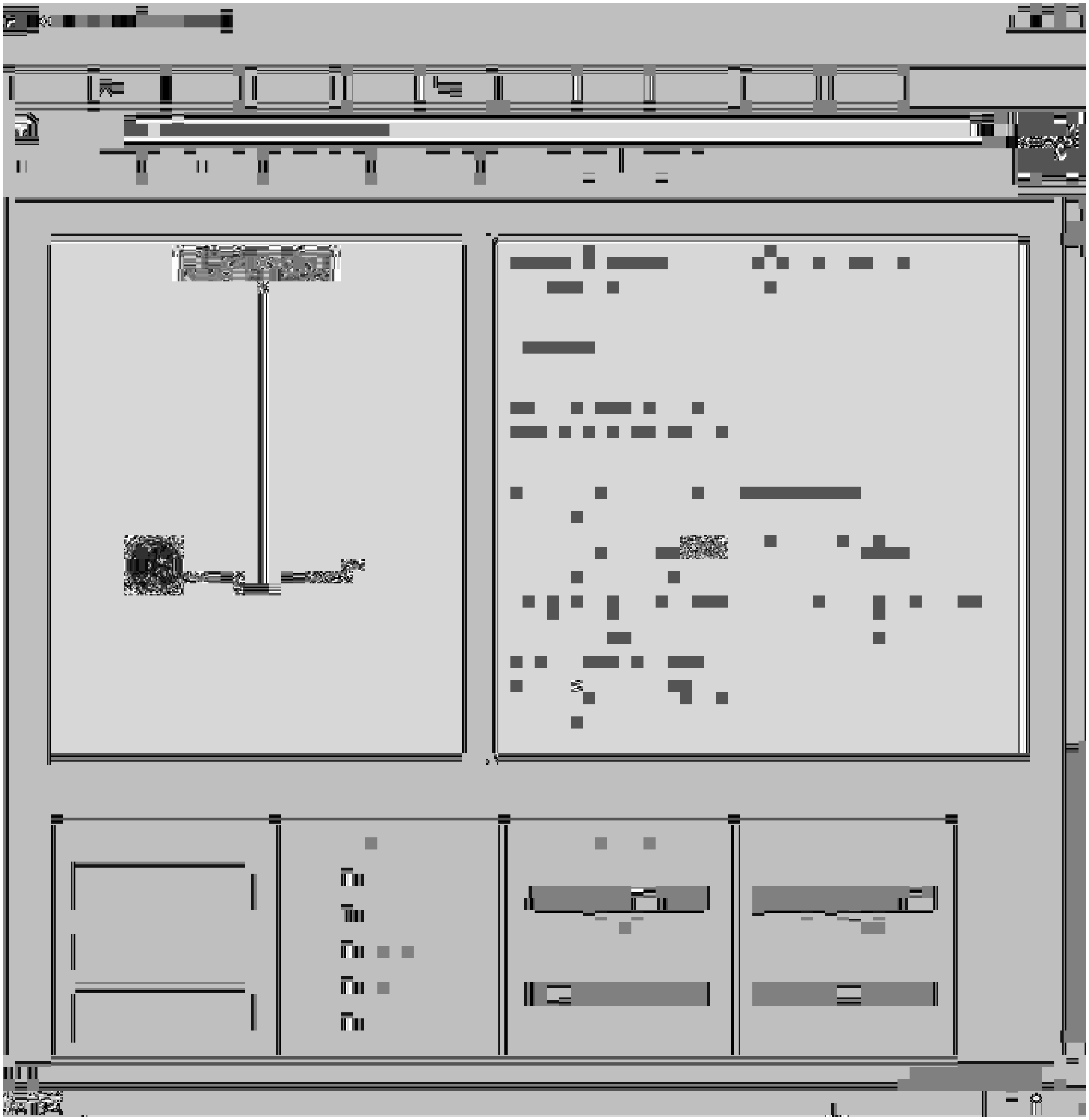

FIGURE 1. PENDULUM SYSTEM