INT. J. COMPUTER INTEGRATED MANUFACTURING, 2000, VOL. 13, NO. 3, 286–297

A modular approach to using computer technology for education and training

STEPHEN ZAHORIAN, WILLIAM SWART, VISHNU LAKDAWALA, JAMES LEATHRUM and OSCAR GONZÁLEZ

Abstract. This paper describes the approach taken to prepare Old Dominion University's undergraduate computer engineering curriculum for technology-based delivery. In order to improve on methods for student learning, technology is now being developed for use in both the classroom and for distance education. To accomplish this, the curriculum content is organized into learning modules that are more fine-grained 'chunks' of learning materials than a three-credit college course. By carefully designing these learning modules—ensuring, for example, well-defined learning objectives, a precedence relationship with other modules, assessment measures, and notational and structural consistency among modules—the modules can easily be reorganized to satisfy a variety of learning objectives. Once produced, a module can be used in a synchronous environment to support student learning in a traditional context, or be part of an asynchronous delivery system such as the Web. The key advantage of this modular structure lies in its flexibility. The investment made to produce the modules may now be recouped by using the modules in both on-campus and distance learning degree and non-degree activities, or by using the modules for performance support. Other advantages include the ability to easily update information in the curricula and the ability to use the best experts for a specific area. In this paper, this modular approach is described in more detail as applied to an undergraduate computer engineering programme.

1. Introduction

This paper reports on the steps taken by the computer-engineering department at Old Dominion University, Virginia, USA, to improve the educational process using technology-assisted education. What are some of the reasons to choose a technology-based approach? First, there is the ongoing commitment to improve the learning environment. Faculties are already making extensive use of course Web pages to help supplement classroom materials. Another important factor is Old Dominion's commitment to delivering quality distance education programmes throughout the state of Virginia and beyond. Since computer-engineering programmes are in high demand because of their close ties to the information technology explosion, Old Dominion would naturally like to offer its undergraduate computer-engineering programme to distance students. Another consideration is the large number of part-time students attending our current campus classes but who often have difficulty coming to campus two to three times per week to attend classes.

For the reasons just mentioned, the faculty in the Department of Electrical and Computer Engineering at Old Dominion University has begun the cooperative development of a technology-based computer-engineering curriculum. Since our faculty size and student body are not large enough to support separate programmes for off-campus and distance students, we began with the assumption that the same technology should be used to improve the educational process for both on campus and off campus students. Rather than use a synchronous TV-based approach for our programme, which is presently the dominant mode at Old Dominion for delivery of distance education (Savage *et al.* 1998), the decision was made to focus on computer-based materials, which can be accessed over the Web. The apparent advantages of this approach as opposed to TV delivery include:

- the ability to deliver material both synchronously and asynchronously;
- the elimination of the need for specialized

Authors: Stephen Zahorian, Vishnu Lakdawala, James Leathrum and Oscar González, Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, Virginia, 23529, USA. William Swart, College of Engineering and Technology, Old Dominion University, Norfolk, Virginia 23529, USA. e-mail: szahoria@odu.edu; wswart@odu.edu

International Journal of Computer Integrated Manufacturing
ISSN 0951-192X print/ISSN 1362-3052 online © 2000 Taylor & Francis Ltd
http://www.tandf.co.uk/journals

receiver sites;
- the ability to have much higher bandwidth display devices (i.e. computer monitors rather than TV screens); and,
- the ability to incorporate computer simulations and interactive exercises as part of the material.

Our previous work involved creating limited Web-page support materials for use with individual courses. Even this modest effort was very time consuming. We understood that the creation of a truly integrated technology-based programme would be an immense task. Therefore, before proceeding with the work, we first carefully assessed the possible impediments and potential benefits from a pedagogical point of view. Given the rapid growth in computer hardware and software, we are confident that our success will not be limited by technology, but only by our proper use of it. Accordingly, we considered the biggest challenges in developing a computer-engineering programme to be:

- keeping the learner engaged;
- persuading faculty to utilize materials developed by others;
- maintaining programme consistency while leaving room for individuality;
- preserving the design and laboratory components of the curriculum;
- managing/facilitating communications among geographically separated students and their faculties;
- integrating the technological components with the current educational infrastructure, including text books; and,
- ensuring reliability of hardware/software.

The inherent advantages of a properly designed computer-based approach to learning include:

- reconfigurable and highly tailorable education and training;
- the ability to provide flexible scheduling for non-traditional students; and,
- the ready availability of archived and background materials for review.

Long-term goals for this undertaking are:

- to improve learning efficiency for both students and teachers;
- to make learning more meaningful and to improve motivation by encouraging students to cooperate with the faculty in planning their programme of study;
- to make the learning process more engaging and enjoyable;
- to make the learning environment more comparable to the information technology world that most of our graduates will soon work in;
- to reduce information overload by focusing on how to access/learn and manipulate/process information rather than on memorizing it;
- to facilitate just-in-time learning and continuous learning every day in the workforce;
- to better serve a wider variety of learning styles; and,
- to integrate more applications into learning materials.

After considering these factors, especially the potential to improve significantly the learning process for our students, a technology-based approach to redefining our curriculum was chosen. In order to derive maximum usability from the large investment needed to create this programme, a modular approach was selected for developing educational materials. Modules are smaller than typical courses and are designed to be reusable and easily configurable to allow the rapid creation of a variety of educational options to satisfy a variety of student needs.

In the remainder of this paper, we first present some background material from the literature on using technology for education. We follow this with a section on general principles, structure, and techniques for creating educational modules. After this are sections that apply the modular approach to portions of our computer-engineering curriculum. We then summarize with our main conclusions.

2. Background

2.1. *Related literature*

One of the important characteristics of the proposed curriculum is the use of small learning units called modules to facilitate the systematic integration of technology-based tools. Although the general idea of modularization of courses and curricula is not new, the integrated approach for the modularization of an entire undergraduate engineering programme has not yet been reported in the literature. Curriculum modularization can lead to fundamental changes in engineering education. For example, Moussavi (1997) proposed a complete modularization of the mathematics instruction, and introduction of these modules just-in-time, as opposed to the traditional approach of teaching the mathematics courses prior to the engineering ones. By completely modularizing an engineering programme, as we propose to do, a more significant change is possible. The resulting programme can then be more properly viewed as a collection of modules rather than a collection of courses. In a completely modularized programme, curriculum changes, such as those suggested by Moussavi (1997), can be more easily incorporated. Students can then play a larger role in managing their curriculum and can earn credit at the module level. Our modularization plan is similar to the general-purpose curriculum redesign in Crynes (1996) where it is observed that modularization is easiest if information technology is used in the delivery of instruction. The advantages of curriculum modularization have also been recognized by the corporate and government training industry. The American Society for Training and Development held an international meeting in 1992 that specifically addressed modular training systems and strategies.

The combination of computer and multimedia tools and Web-based delivery can form an excellent complement to a live or online course. Examples of the former possibility are discussed in Haile (1998) where the initial experiences of enhancing and supporting classroom instruction using technology at Hofstra University are presented and discussed. Their pedagogical goals and procedures are also applicable in engineering education. There are many examples of online courses using computer and multimedia tools to some extent. An interesting view of where this is leading is presented in Van-Dusen (1997), which describes and analyses a possible technology-based virtual campus.

One of the primary reasons to emphasize computer tools is to improve the students' understanding of a topic by actively engaging them in activities such as computer simulations and computer interactive problems. Two examples of this development are presented in Hoover and Abhaya (1995) and in Yost *et al.* (1997). In Hoover and Abhaya (1995) the results of the NSF Center for Interfacial Engineering Curriculum Development Project are presented. This Center brought together engineers and instructional design educators to design eight computer-based instructional modules. The development of computer-based instructional modules integrated with the curriculum of an undergraduate control systems course is presented by Yost *et al.* (1997). Another reason for using computer and multimedia tools is to make the learning of engineering more relevant by integrating education with practical problems. In Griffith *et al.* (1997), the development of multimedia modules by the Manufacturing Engineering Education Partnership (MEEP)—-which complement lectures and laboratories—is presented. The multimedia modules developed in Woolf *et al.* (1997) teach manufacturing design. In Furman (1996), the development of Web-based modules that integrate the theory and practice of mechanical engineering is presented.

2.2. *Reported issues*

In the literature of technology-based education, there is a consistent list of issues, concerns and recommendations. The important issues can be classified into three groups: instructional design, faculty time management, and technology management. The most important issue is instructional design, which consists of several important sub-components. The core difficulty is that the information in a multimedia-based module cannot be modified on the fly as is typically done in a blackboard-based course. Thus, careful planning is needed to ensure that these modifications will not be necessary. Several authors have noted that a direct transcription of lecture notes into an electronic format will not lead to a successful educational experience for most students. This information might be useful as a course supplement but for quality education special attention must be paid to the variety of student learning types. It has been conjectured, however, that a well-planned set of multimedia instructional units can be designed to appeal to most student learning types. In addition, technology-based education can make it possible to improve the education since it makes it simpler to implement alternative instructional approaches, as presented by Berge (1999). The instructional modules can now be easily designed to be student-centred as opposed to teacher-centred, where most of the information must flow from the instructor and reading material into the student. Furthermore, it does not appear that modules should be completely self-paced, although some components of instructor-paced mastery courses (Wankat and Oreovicz 1993) are useful to accommodate the time constraints of the more diverse distance learners.

The next important issue is that of faculty time management. There is complete agreement that the development of multimedia learning modules is a time-intensive process. More surprising are the reports by Forinash *et al.* (1998) that faculty time requirements during the course also often increase due to additional communications with students, much of which happens at night and on weekends. Suggestions to handle this problem are careful time management and delegation via discussion groups so that students help each other. Another way to reduce the time needed to read and process student communications is to request the use of specific keywords in the e-mail subject lines and to impose software standards and naming conventions for submitted files.

The final important issue is that of technology management. Students are now more aware of new technologies and there is little need to teach them to use e-mail and the World Wide Web, but certain standards should be developed. Tutorials should still be provided for more specialized technological resources. It is also important to be prepared for the failure of the technology. It is possible for Web or e-mail or discussion group servers to fail at the worst possible time. Another technology problem that can be a 'time sink' is the use of incompatible software packages or even different versions of the same programme. This can be resolved with the requirement of a standard set of software packages across the curriculum.

There is still no final verdict on the educational benefit of technology-based instructional modules. In Kadiyala and Crynes (1998), and Coleman *et al.* (1998) some of the current research results are detailed. The bottom line is: the use of technology does appear to improve the learning environment but the research is not complete. Success or failure depends very much on the details of implementation. It is too early for detailed cost/ benefit analyses to be available.

3. Methodology

3.1. *The learning module concept*

In order to capitalize on the large investment of time required to create computerized materials for web-based delivery, our approach is based on small 'building blocks' of educational materials, referred to as learning modules. These modules are loosely defined as the smallest autonomous 'units' of educational material that cover a specific concept. Typically, modules are much smaller than traditional courses. Each module is characterized by attributes including:

- a specific precedence relationship with related modules;
- specific learning objectives associated with each module; and,
- assessment measures to determine whether or not learning objectives have been met.

The key advantage of this modular structure lies in its flexibility. The original investment made in producing modules may be recouped by the repeated use of the modules for both on-campus and distance learning activities. The modules can be used either as part of a technology-assisted course or they may be made available for specialized needs. Students or professionals needing to refresh knowledge or to support performance of a task being undertaken can also access the modules upon demand. Furthermore, an advantage is that a modular architecture provides a framework for streamlined information updating. In addition, the best expert or experts for a specific area can be used to develop specialized modules.

Once the modular architecture of the curriculum is defined, the development of individual modules begins. As per Crynes and Hawley (1995), the principles guiding development are that modules should:

- be electronically storable and deliverable;
- incorporate knowledge from the best sources available while ensuring that different perspectives and formats appeal to multiple learning styles;
- incorporate appropriate multimedia, hypermedia, simulations, and collaborative and synthetic environments as appropriate to facilitate learning;
- present materials from a synthesized multidisciplinary context to facilitate re-use and multiple use;
- provide synchronous and asynchronous electronic interactivity with the instructor(s), fellow students, and external experts;
- have links with related URL's, including digital libraries;
- incorporate, as a rule, significantly less content than a one-semester course;
- incorporate learning diagnostics for student self-assessment;
- incorporate student feedback and performance records for continuous module improvement;
- be competency based (e.g. have well-defined learning objectives and appropriate assessment methodology);
- be interesting and motivating; and,
- clearly state prerequisite skills, knowledge, abilities and attitudes.

In the remaining parts of this section, we will outline our strategy to attempt to meet these principles.

3.2. *Formal module definition*

A uniform module structure not only facilitates the development process, but also provides students with a consistent format. This type of formalism is provided for the module interface, interconnection, and internal structure while leaving freedom for individualized content development.

3.2.1. *Module interface.* Individual modules are defined at the interface level by:

- the key learning objectives, in terms of specific knowledge areas;
- the pre-requisites by topic (with the associated module where available);
- the expected outcomes, in terms of applying and using learning objectives;
- time and difficulty weighting factors to complete the module; and,
- typical subsequent modules.

Using a common interface structure assists in the interconnection of the modules and identification of internal content. The input/output descriptions are sufficiently detailed to allow reconfiguration of modules into a variety of individually designed 'courses'.

3.2.2. *Module interconnect.* The relationship between modules is defined graphically to assist both students and the faculty in visualizing the curriculum and in designing 'courses' (see figure 1). While backward links refer to prerequisite modules, forward links refer to typical subsequent modules. The diagram includes a broader description of module content and objectives, and estimates the typical number of hours for completion. Hyperlinks may provide expanded module descriptions.

3.2.3. *Internal structure.* Goals for defining the internal structure of the module are to provide:

- some general consistency in content between instructors;
- core material from which an instructor may tailor his or her own presentation while allowing each to capitalize on his or her own individual style of presentation; and,
- tools and demonstrations to supplement the material.

The resources provided in the module, which assist in achieving these goals, are summarized below.

Detailed module content outline. Module content is organized into a prioritized outline that identifies the principle concepts to be covered and allows instructors some discretion in presenting less important concepts.

Basic information. This is a mixture of text, figures, video, sound, and charts, conveying the key concepts. In order to engage the asynchronous learner, however, the material is heavily interspersed with short questions and workbook type entries requiring completion in order to continue.

Simulations and demonstrations. Whenever possible, simulations and animated demonstrations are used to illustrate concepts. The timing of control, address and data on a computer bus is an example of material ideally suited to an animated demonstration. These simulations typically employ user control over the simulation. The simulations offer added explanations and pose questions.

Interactive exercises. Here the student will complete exercises independently, except for monitoring and guidance from the system. Integrated problem-solving walks students through the thought process. First, an overview of the steps is provided with an accompanying explanation. The student participates to varying degrees in the solution, but retains access to the instructor's solutions.

Assessment tools. These are discussed in more detail in section 3.3.

3.2.4. *Classes of modules.* Modules fall into three classes: informational, integration, and assessment. Most learning modules are informational, incorporating general learning with a limited assessment. They focus on a specific concept and may draw material from other modules to develop a concept of interest. Integration and assessment modules are tailored to specific needs at various points in the curriculum to perform activities related to a collection of informational modules. Integration modules combine concepts from several modules. While the structure, content, and activities of the integration module are similar to those of an informational module, the module is defined as a different class to indicate the integration purpose of the module. Integration is an important concept in engineering that must have special attention. Occasionally, it is desirable to assess the knowledge attained from a set of modules—thus requiring an assessment module. These assessment modules might also include review material to assist in studying, using appropriate links back to the original material in other modules.

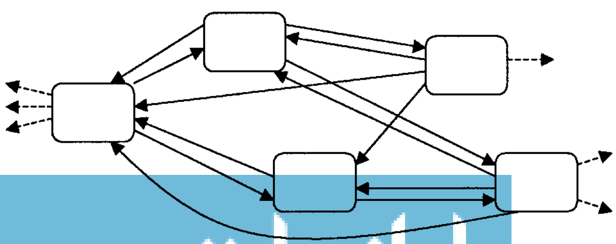

Figure 1. Module interconnection diagram.

3.3. *Assessment methods*

Although assessment may be even more critical for technology-based learning as opposed to learning by traditional methods, technology can also provide convenient, frequent, and thorough assessment. This would include confidence building self-assessment as well as more conventional testing designed to certify mastery of concepts. Additionally, programme assessment will help to determine the degree to which modules are satisfactory for facilitating learning. Note that this last type of assessment is a fundamental component of the ABET 2000 (Accreditation Board for Engineering and Technology 1999–2000) engineering accreditation criteria.

The following tools will assist in accomplishing assessment.

- *Monitoring of log on time for each module/exercise/problem.* This automatic measure monitors how much time students spend on each module, and how that time is spent. Statistics will be used to update the expected completion time for each module and identify difficult informational segments needing revision.
- *'Easy' short answer questions interspersed throughout informational material.* These questions will assist in student self-assessment as well as attempt to engage the learner in a manner that mimics classroom interaction. These untimed questions, with unlimited attempts, must be answered correctly before continuing. Response parameters will help determine the difficulty levels of the modules.
- *Homework exercises.* These will be patterned after the computerized homework used, for example, by Saxena (1998) at Purdue University in their Physics Online System. Automated techniques will provide a range of parameter values, and customized problems, with immediate feedback to the student.
- *Module exams.* The primary measure of assessing module mastery will be the timed exam with automated test banks that will vary parameter values and randomize questions to ensure testing integrity.
- *Comprehensive 'milestone' exams.* Since students have flexibility in tailoring their programmes, these exams will be used to ensure that core concepts are mastered. More importantly, these questions will typically draw on material from several modules to determine if concepts from these various modules are sufficiently integrated in the student's mind. These milestone exams also could be used to certify experiential learning, evaluate college transfer credits and AP courses from high school, and act as an exit exam for the overall programme.
- *Review exams.* These exams will test basic concepts, help students review past material, and help students prepare for an upcoming module.

Preparation of large computer-scoreable exam banks (a tremendous effort) is critical to the success of the programme. Poor student performance will trigger the need for an in-depth conversation with the module instructor.

3.4. *Implementation issues*

The development process involves an iterative cycle of development, field-testing, and revision of modules for both on campus and off campus students following the process outlined below:

- divide curriculum content into segments each focusing on a specific concept;
- develop a detailed content outline for each module;
- develop and integrate informational content, examples, problems, and multimedia exercises for each module, taking care to maximize interactive components;
- field test modules with both on and off campus students; and,
- use student feedback to revise each module.

4. The undergraduate Computer Engineering programme at Old Dominion University

4.1. *The overall programme*

The Computer Engineering curriculum at Old Dominion University consists of 124 total credits. Of these, 58 credits form the computer-engineering core including 21 credits from the computer science core. Figure 2 is a flow chart of the programme. The major core credit distribution is as follows.

Required ECE lecture courses	28 Credits
Required ECE laboratory courses	4 Credits
Senior design project	5 Credits
Required CS lecture courses	21 Credits

4.2. *Circuit theory modules*

Tables 1 and 2 show the manner in which the instructors of the courses decided to break down circuit theory, ECE 201 and ECE 202, into learning modules, based upon pedagogical considerations. The fact that there are six modules associated with each of the two courses is coincidental. Figure 3(*a*) shows the prerequisite relationships between each of the modules that collectively represent ECE 201 while figure 3(*b*) shows the same for ECE 202, a course that lists ECE 201 as a prerequisite. The links between figure 3(*a*) and figure 3(*b*) indicate the only direct prerequisite relationships between the two courses.

Figure 3, when compared with figure 2, illustrates the difficulty of a traditionally structured curriculum to accommodate specific learning requirements of a non-traditional student. For example, suppose a student with some electronics technical background requires knowledge to analyse RLC filters with Laplace Transforms. Since this information is contained in two modules of ECE 202 (MM2-MM3), and since ECE 201 is a prerequisite for ECE 202, the student would not be allowed to register for the desired course (ECE 202) until she/he completed ECE 201. Thus, in order to gain the knowledge required now, the student would have to spend one full semester completing the prerequisite. Furthermore, the student would have to pay the tuition for a course containing no material required to learn

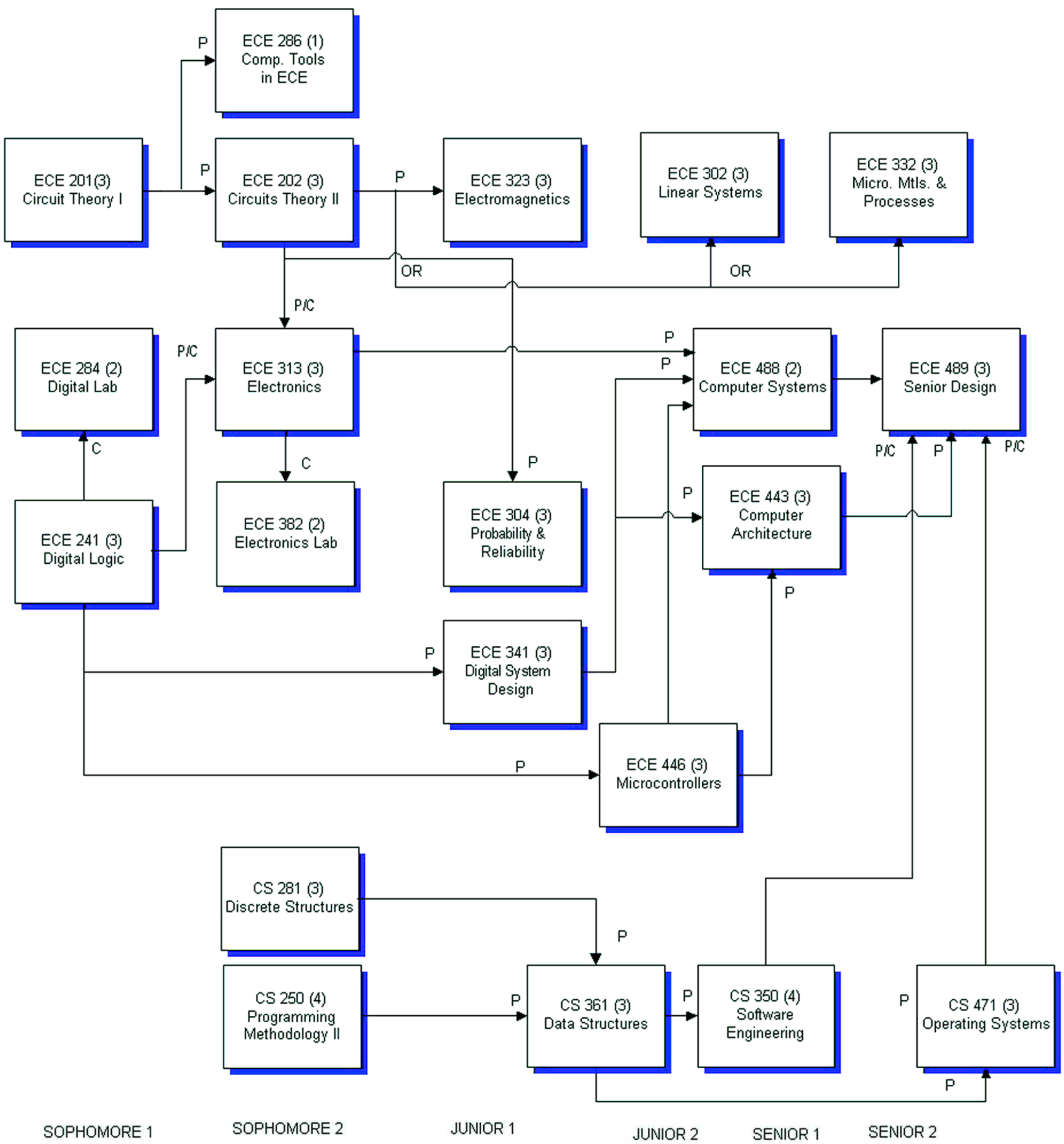

Figure 2. Flow chart for the computer engineering core curriculum.

Table 1. Modular breakdown of ECE 201.

Module	Topic
M1	Circuit component and connection laws.
M2	Formulation and solution of network equations.
M3	Network reduction techniques and network theorems.
M4	Modelling of two-port networks.
M5	Electrical signals and energy storage elements.
M6	Analysis of first order networks.

Table 2. Modular breakdown of ECE 202.

Module	Topic
MM1	Analysis of second order networks.
MM2	Laplace transform theory.
MM3	Laplace transform analysis and frequency response of circuits.
MM4	A.C. steady state phasor analysis.
MM5	A.C. steady state power.
MM6	Two-port electrical filters and frequency domain analysis of networks.

Laplace Transform Theory and its applications to circuit analysis. It could be argued that, for a traditional student pursuing a degree full time with no other responsibilities, this is not particularly important. However, for a part-time student working in a technical position, the need to know something before it is covered in the curriculum may be very real.

Table 3 lists a number of individualized 'learning paths' that a modularized curriculum would make available to learners using only the circuit theory modules. Instead of having to take all subject matter contained in ECE 201 before learning any of the subject matter contained in ECE 202, learners can select the 'shortest' path to the material they require. Learners pursuing a degree would eventually be required to complete all modules. However, the sequence in which modules are taken is tailored to individual requirements. Each learning path only needs to be followed until the desired module is reached. The learning path of row 6 in table 3 indicates that two independent paths (M1, M2 and MM2 must be completed before module MM3 can be taken).

By developing a modular architecture for the curriculum, wherein each module has specified learning results and associated assessment methodology, the modules can then be used for multiple purposes, including professional development/ short courses and performance support tools. Thus, in addition to making materials more accessible so that learning can take place closer to the time it is required, modularization can reduce the overall costs associated with meeting multiple requirements.

4.3. *Module example: computer architecture controller design*

This section gives a detailed example on controller design in computer architectures to demonstrate the structure of a module. The main function of a controller is to use the current state of the architecture and the current instruction to generate a sequence of control signals that dictate how the architecture executes an instruction. Controller design is classically taught in a computer architecture class after the architecture structure is presented. This example of a module for controller design demonstrates some of the general concepts previously presented.

4.3.1. *Module interface.* The first step in module development is to define the interface, as documented in table 4. Module content is derived from the interface description. Note that this module completes a larger topic of basic computer architecture design. It is therefore appropriate to have an assessment module after completion of this module, or at completion of the integration module. This assessment module is related to the midterm exam in a conventional computer architecture course.

4.3.2. *Internal structure.* The first task is to provide an expanded version of the outline for controller design as given in the module interface. If the content material is drawn only from this detailed outline, consistency among multiple offerings of the module will be ensured.

The base material for the example module includes the following;

PowerPoint lectures—Lectures include animation to demonstrate concepts such as the sequencing of control signals. The lectures may also be integrated with software tools that can be custom made or built from off-the-shelf packages such as Microsoft's NetShow or Lotus' LearningSpace. These packages can be used to combine slides with video, audio and Web links.

Java simulations—Java simulations allow students to interact with a controller to understand its operation better. A hardwired controller simulation is shown in figure 4. Students have the option to define any instruction for the given architecture. For each instruction, the student defines the RTL for the instruction and the control signals required for each clock cycle. She then compares her computed control equations to the simulation's equations to verify her design. Students then single step through an instruction to observe the functionality of the controller architecture on each clock cycle, thus observing the relationship between the control signals and the RTL.

Design examples. Engineering students must understand computer architecture well enough so that they can also design new architectures. It is crucial to reinforce the ideas needed for the design process. The ability of students to complete designs successfully,

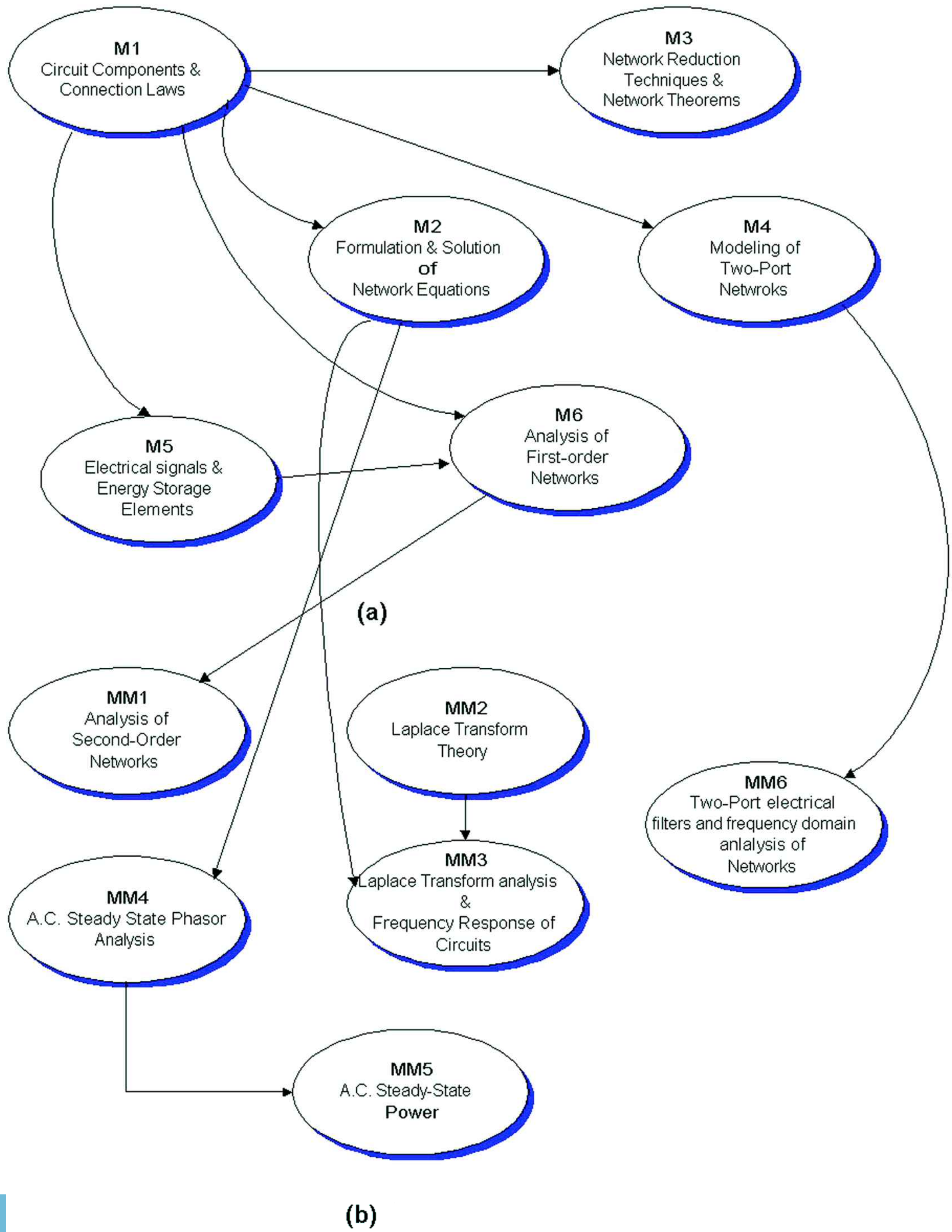

Figure 3. Structure of modular relationship for (*a*) ECE 201: Circuit Theory I and (*b*) ECE 202: Circuit Theory II.

particularly when they are primarily learning this material in an asynchronous mode from Web-based technology, will be a critical measure of the success of the programme.

4.3.3. *Assessment.* Assessment in this module comprises homework, testing, and the completion of a component of a large project. An infrastructure for homework assignments, such as that used in Saxena (1998), facilitates the learning experience over the Web. Homework will be augmented with a simulator for each type of controller (hardwired and microprogrammed) to demonstrate understanding of a base concept. For example, in hardwired controller design, control signals are defined by a control equation driven by underlying hardware. The simulator for this assignment would provide the student with a definition of an architecture and instructions for that architecture. The student would then derive selected control equations and enter them into the simulator. He could then observe the behaviour of the system with his equations. When satisfied, he automatically submits his solution.

Computer architecture is primarily a design concept. To address this need, a very large project involving the design and simulation of the central processing unit of an architecture is undertaken by the students. In a conventional classroom, the project is broken into phases, each phase associated with a specific topic in the class. Students work on the phase when the topic is completed. In the module structure, these phases now become part of the assessment process in individual modules. In this sample module, the student will have already developed the design of the architecture in the form of the required components, data paths, and how

Table 3. Distinct learning paths within ECE 201 and ECE 202.

M1, M3
M1, M5
M1, M4, MM6
M1, M5, M6, MM1
M1, M2, MM4, MM5
M1, M2/ MM2, MM3

Table 4. Computer architecture controller design module.

Learning objectives

The purpose of this module is to learn the process for design of the control unit for a CPU. Given a macroinstruction and current system state, the controller generates the sequence of control signals or microinstructions that produce the activities to execute the macroinstruction. The student will learn:

- how to identify which control signals are necessary for each macroinstruction in the register transfer language (RTL)
- hardwired controller design
- microprogrammed controller design
- horizontal microprogramming
- vertical microprogramming

Prerequisite information:

Information	Module
1. Register Transfer Language (RTL)	Register Transfer Language
2. Computer Macroinstructions	Instruction Set Design
3. Architecture Components	Computer Architecture Design
4. Data Paths	Computer Architecture Design
5. Architecture/ Controller Modelling	Architecture/ Controller Modelling
6. Boolean Algebra	Boolean Algebra
7. 2-Level Logic	Combinational Logic
8. Registers (Counters, Shift Registers)	Sequential Circuit Components
9. Read Only Memory (ROM)	Programmable Logic Devices

Expected outcomes

The student will be capable of designing and implementing a computer controller. To demonstrate this capability, the student will be expected to pass a test and design the controller for his computer architecture project. This module completes the basics of computer architecture design and will require a following assessment module to evaluate the integration of these concepts.

Time and difficulty weighting factors

TBD

Typical subsequent module

- Computer architecture assessment
- Computer architecture project
- Pipeline design
- Memory hierarchy overview

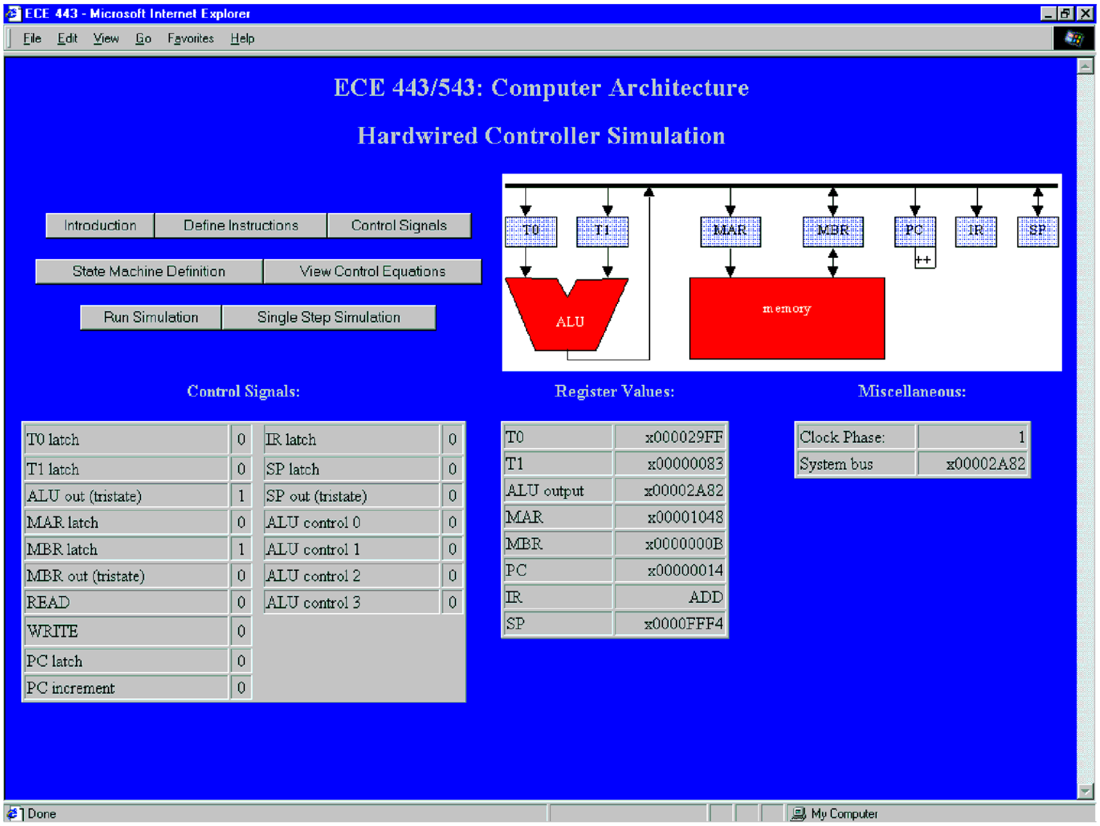

Figure 4. Java simulation screen capture for hardwired controller design.

instructions are to execute on the architecture. From this, the student must now design a controller to support his design. The instructor provides feedback on the design, but the project component is not officially graded. This is because design is an iterative process requiring the student to revisit previous work. Final assessment is not done until a later module, which ties all of the components together. This final module is the Computer Architecture Project Module (an integration module), where the students simulate their designs and document the design and results. Only at this time can final assessment be completed.

5. Conclusions

The success of this 'work in progress' depends on many factors, beginning with the modularization of the computer engineering programme with well-defined learning objectives for each module. Just as important is the proper integration of modules in the overall programme with appropriate prerequisites specified. This time-consuming effort will require the close collaboration of many faculties. Another important step is the creation of instructional content for the modules, using the module templates as introduced in section 3. It is essential that modules be effective with a variety of student learners; that appropriate interactivity is added with computer simulations and interactive problems; that appropriate communication channels are established such as discussion groups, video-conferencing and e-mail; that assessment with appropriate feedback is made an integral component of the module; and that, overall, the best instructional techniques are used to accomplish the educational objectives. All of these issues require that the engineering faculty work with professional educators and instructional designers.

One of the real advantages of using technology-based education lies in the ability of students to access additional information on a topic via Web links to information in past modules, Web links to other resources (potentially Web-based textbooks), and primary and secondary references (textbooks, workbooks, etc). We are confident that the strategies presented in this paper will result in improvements in engineering educational systems. These methods can also be readily adapted to workforce training for non-degree activities.

References

Accreditation Board for Engineering and Technology, 1999–2000, Criteria For Accrediting Engineering Programs, Inc., 111 Market Place, Suite 1050, Baltimore, MD 21202, http://www.abet.org/eac/EAC_99-00_Criteria.htm.

Berge, Z. L., 1999, Interaction in post-secondary Web-Based learning. *Educational Technology*, **39**(1), 5–11.

Coleman, J. N., Kinniment, D. J., Burns, F. P., Butler, T. J., and Koelmans, A. M., 1998, Effectiveness of computer aided learning as a direct replacement for lecturing in degree-level electronics. *IEEE Transactions on Education*, **41**(3), 177–184.

Crynes, B.L., 1996, Curriculum Granularity. Proceedings of the 26th Annual Frontiers in Education Conference: Teaching and Learning in an Era of Change, Salt Lace City, UT.

Crynes, B. L., and Hawley, J. A., 1995, Professor silicon and professor maestro—the perfect combination. *Proceedings of the 25th Frontiers in Education Conference*, Atlanta, Georgia, http://fairway.ecn.purdue.edu/~fie/fie95/2a3/2a33/2a33.htm.

Forinash, K., Rumsey, W., and Wisman, R., 1998, Interactive and collaborative uses of the Web. *Enhancements: How Using Technology Changes What Faculty Do*, IndianaHigherEducationTelecommunicationSystem, http://www.ihets.org/distance_ed/fdpapers/1998/ index.html.

Furman, B. J., 1996, The Un-Lecture: a computer assisted curriculum delivery approach for the effective teaching of mechanical design. *Proceedings of the 26th Annual Frontiers in Education Conference: Teaching and Learning in an Era of Change*, Salt Lake City, UT (New York: IEEE Press) pp. 1379–1382.

Griffith, M. L., Lamancusa, J. S., Jorgensen, J. E., and Velez, J., 1997, Multimedia courseware to enhance the classroom experience. *Proceedings of the 27th Annual Frontiers in Education Conference: Teaching and Learning in an Era of Change*, Pittsburgh, PA (Champaign: Stipes Publishing) pp. 1171–1174.

Haile, P., 1998, Multimedia instruction initiative: building faculty competence. *The American Association of Colleges for Teacher Education Annual Meeting*, New Orleans, LA, 25–28 February, EDRS Report ED 418 056.

Hoover, S. J., and Abhaya, P. S., 1995, Instructional design theory and scientific content for higher education. *Proceedings of the 1995 Annual Convention of the Association for Educational Communications and Technology*, Anaheim, CA, EDRS Report ED383302.

Kadiyala, M., and Crynes, B., 1998, Where's the proof? A review of literature on effectiveness of information technology in education. *Proceedings of the 28th Annual Frontiers in Education Conference: Moving from 'Teacher-Centered' to 'Learner-Centered' Education*, Tempe, AZ (Piscataway: IEEE Press) pp. 33–37.

Moussavi, M., 1997, Modular curriculum: an alternative approach to engineering education. *Proceedings of the 27th Annual Frontiers in Education Conference: Teaching and Learning in an Era of Change*, Pittsburgh, PA (Champaign: Stipes Publishing) p.704.

Savage, A., Stanley. W., and Swart, W., 1998, A ten year perspective on delivering undergraduate engineering technology programs via distance learning at Old Dominion university. *Proceedings of the 1998 National Meeting of the Accrediting Board for Engineering and Technology*, October.

Saxena, V. K. , Nakanishi, H., Shibata, E. I., Elmore, D., and Bortoletto, D., 1998, CHIP: computerized homework in physics. *Enhancements: How Using Technology Changes What Faculty Do*, Indiana Higher Education Telecommunication-System, http://www.ihets.org/distance_ed/fdpapers/1998/index.html.

Wankat, P., and Oreovicz, F., 1993, *Teaching Engineering* (McGraw-Hill).

Woolf, B. P., Poli, C., and Grosse, I., 1997, Multimedia systems and intelligent tutors for teaching design for manufacturing. *Proceedings of the 27th Annual Frontiers in Education Conference: Teaching and Learning in an Era of Change*, Pittsburgh, PA (Champaign: Stipes Publishing) pp. 711–716.

Yost, S. A., Al-Holou, N., and Hideg, L .M., 1997, Development of computer based instruction in control systems curriculum. *Proceedings of the 27th Annual Frontiers in Education Conference: Teaching and Learning in an Era of Change*, Pittsburgh, PA (Champaign: Stipes Publishing) pp. 671–675.

Van Dusen, G. C., 1997, *The Virtual Campus: Technology and Reform in Higher Education*, ASHE-ERIC Higher Education Report, 25 (5), EDRS Report ED412816.