AC 2007-1965: UNDERGRADUATE EMBEDDED SYSTEM EDUCATION USING ADAPTIVE LEARNING TECHNOLOGY

Liang Hong, Tennessee State University

Liang Hong received the B.S. degree and M.S. degree from Southeast University, Nanjing, China, in 1994 and 1997, respectively, and the PhD degree from University of Missouri, Columbia, MO, in 2002, all in electrical engineering. Since 2003, he has been with Tennessee State University, Nashville, TN, as an Assistant Professor in the Department of Electrical and Computer Engineering. He has held summer visiting appointment at Vanderbilt University, Nashville, TN, in 2006. His research interests include digital communications and multimedia signal processing with a recent focus on wireless video transmission, modulation classification, speech enhancement, and sensor networks. He is a member of IEEE and ASEE.

Md Hasanuzzaman, Tennessee State University

Md Hasanuzzaman received the B.S. degree and M.S. degree from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 1996 and 1998, respectively, and the PhD degree from University of Tennessee, Knoxville, TN in 2004, all in electrical engineering. Since 2005, he has been with Tennessee State University, Nashville, TN, as an Assistant Professor in the Department of Electrical and Computer Engineering. He was a research assistant of Oak Ridge National Laboratory and worked on advanced microprocessor design at IBM, Austin, TX research facilities. He has also held summer visiting appointment at Vanderbilt University, Nashville, TN, in 2006. His research interests include advanced electronics, high temperature electronics, analog and digital circuit design, computer architecture, and embedded system design. He is a member of IEEE and ASEE.

Undergraduate Embedded System Education Using Adaptive Learning Technology

Liang Hong Md Hasanuzzaman

Department of Electrical and Computer Engineering
Tennessee State University
3500 John A Merritt Blvd, Nashville, TN 37209

Abstract

Embedded systems are the fastest growing areas of computing in recent years. This growing popularity calls for engineers with experience in designing and implementing embedded systems. This paper describes an undergraduate embedded system design course in our curriculum that is offered as technical elective for senior students. The course contents, organization of the materials and the laboratory are presented. This course is designed by introducing a balanced view of software and hardware concepts in the design process. The objectives of this course are to expose students to the field of embedded systems with knowledge foundation, and to provide them with hands-on experience via a sequence of laboratories. A student-centered adaptive blended learning technology based on the Courseware Authoring and Packaging Environment (CAPE), and the experimental Learning Management System (eLMS) is also presented in the paper. Comparing with other course management software, such as WebCT and BlackBoard, this web-based adaptive blended learning technology provides a better intelligent tutoring and learning environment. It provides an opportunity for educators to assess the strengths and weaknesses of each student and provide adapting learning activities in response. It can significantly improve the student retention by providing them with immediate feedback after the quiz. It can also be integrated to WebCT and BlackBoard to enhance the efficiency of knowledge delivery. Furthermore, the web-based CAPE/eLMS adaptive learning environment will not only enable adaptive in-class instruction, but also enable powerful possibilities for adaptive learning activities outside the classroom, both in preparation for in-class activities and in following them up. The CAPE model introduced in this paper offers a generalized framework and can be easily modified by educators to fit their needs.

Introduction

Embedded systems are information processing systems embedded into enclosing products such as cars, telecommunication or fabrication equipment. Real-time constraints, dependability and efficiency are common characteristics of these systems [1]. Embedded systems are the fastest growing areas of computing in recent years. Over ninety percents of all processors are used in embedded applications, and almost every facet of modern life, such as cellular phones, TVs, video game consoles, GPS devices, network routers, cars and space shuttles, contains embedded processors [2]. This growing popularity calls for engineers with experience in designing and implementing embedded systems. The educators must make available the necessary skills to their students by incorporating the entire system design and implementation procedures, including specifications and modeling of embedded systems, hardware/software partition and co-design; validation and implementation.

In recent years, computers have become an integral part of our day-to-day lives. The students who enter college today have become accustomed to Web browsers. They access information, make purchases and exchange data on the World Wide Web. Therefore, learning through web-based environments has dramatically increased and is now increasingly influencing the nature of teaching and learning in electrical and computer engineering education [3].

The pedagogical value of web-based educational tools has been demonstrated in undergraduate embedded system education [4-5]. Usually, the educators post course materials such as syllabus, course schedule, lecture notes and homework on their website or deliver them through a virtual learning environment such as WebCT or BlackBoard. These same course materials are presented to all the students who take this course and are fully interpreted by a teacher in the context of the targeted learning situation. However, the different talents and academic goal of students require the educator to address learners as individuals, assessing their strengths and weaknesses and adapting learning activities in response. On the other hand, by examining the effect of immediate feedback, delayed feedback and no feedback on students' performance when confronted with previously encountered quiz questions on the final examination, Brosvic et al. found a significant improvement in retention when students were initially provided with immediate feedback rather than delayed feedback or no feedback, and even greater retention when provided with multiple attempts on the initial encounter [6]. However, most web-based learning technologies including WebCT and BlackBoard are still lack of immediate feedback to the students, since assignments performed outside class must await human evaluation and subsequently be returned to the learner for reflection.

In this paper, an undergraduate embedded system design course that uses the adaptive learning technology to provide efficient education through web-based tools and immediate feedback to the students on their performance is described. The paper is organized in the following manner. First, the student-centered adaptive learning technologies, Courseware Authoring and Packaging Environment (CAPE), and web-based delivery platform, the experimental Learning Management System (eLMS) is briefly introduced. Second, the advantages of using CAPE and eLMS are elaborated by comparing the CAPE/eLMS platform with the widely used course management software WebCT and BlackBoard. Third, the course organization, content areas and the required laboratory are illustrated. Fourth, the CAPE models for the undergraduate embedded system design course are addressed. Finally, conclusions are drawn.

Web-based adaptive learning technology

Adaptive learning technology is referred to "student centered" learning technology. It is one of four fundamental quality aspects of effective learning environments that were recognized by National Science Foundation (NSF) [7]. Adaptive learning technology uses interactions or prior knowledge about an individual learner to dynamically alter the flow or content of learning activities. The earliest adaptive learning technology, intelligent tutoring systems [8] that acquires and responds to knowledge about individual learners, can trace its origins back some 30 years. Recently, the NSF Engineering Research Center for Bioengineering Educational Technologies (called VaNTH) [9] developed a web-based learning infrastructure for adaptive learning. It consists of two primary components: Courseware Authoring and Packaging Environment (CAPE) and the experimental Learning Management System (eLMS).

CAPE [10] is used to design how learning materials are used to create an adaptive learning experience. It is a graphical modeling language. In this language, iconic nodes represent authoring concepts, and edges represent various kinds of relationships among these concepts. The CAPE designs specify when, or under what circumstances, content elements are presented to a learner during the course of a learning experience. Interactive elements can elicit information from a learner, and the outcomes are available immediately to adaptations incorporated into designs. The content and computational elements can be interchanged with traditional development tools. The completed designs can be directly uploaded to the delivery platform for subsequent assignment to learners.

To enable complex representations to be created, CAPE supports hierarchy; i.e., larger definitional units can be built up from smaller units. To enable reusability, CAPE supports abstraction and refinement; i.e., definitional units can be used as the starting point for other definitional units, and the latter "inherit" changes from the former. To assist the educator in creating, previewing, and packaging designs, CAPE also provides a set of extension components including the *ContentImporter* that can be used to automatically build contents, the *ContentPreviewer* that allows the learning contents to be launched in a web browser, the *DeliveryPreviewer* that enables authors to assure that the authoring task has been performed correctly, and the *ContentPackager* that creates the target representation of the courseware artifact and optionally uploads it to the delivery platform.

eLMS [11] is a web-based delivery platform that supports interoperation using web services, both in conjunction with enacting courseware designs and in managing domain-specific objects, such as classes, users, and courseware. The heart of the eLMS platform is a model-based delivery engine that enacts learning designs authored with CAPE. The eLMS platform automatically captures detailed instrumentation of the CAPE design enactments, and additional instrumentation—to support grading using custom rubrics that can be incorporated into courseware designs with CAPE. eLMS allows learners to review materials and activities across multiple sessions, to take private notes that can be exported from the learning environment, and to access context-sensitive help resources provided by learning designs. eLMS instructors and teaching assistants can manage the rosters of classes and make courseware assignments to a class or to individuals in the class. The status of learners completing assignments can be monitored, learners can be selectively released from synchronization points defined by learning designs, and instructors can replay assignments with learners during face-to-face meetings. Courseware revisions uploaded by authors are differentially versioned to avoid disruption of in-progress enactments with learners. The resulting delivery records can be queried by instructors. These capabilities enable an intimate understanding of what learners actually do with on-line learning experiences, which is essential to making incremental improvements over time. Furthermore, when the learner accesses the courseware via eLMS, the eLMS will automatically save the status of the learner's progress. Therefore, the learner can resume his/her study from where they leave.

Figure 1 shows an example of the eLMS delivery interface for the students. Without overly intruding on the instruction, a simple toolbar located along the left bottom of the browser border provides a large amount of interface functionality, such as forward, backward, take personal notes, and so on.

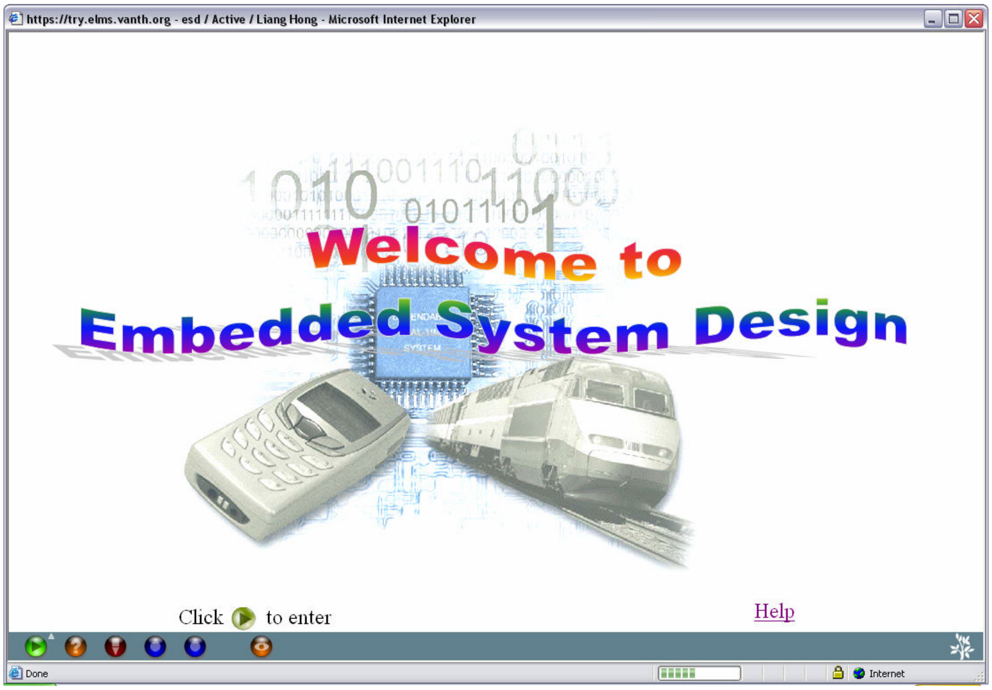

Figure 1. eLMS delivery interface for students

Advantages of the CAPE/eLMS platform

Adaptive learning uses what is known about an individual learner, *a priori* or through interactions, to dynamically alter the flow or content of learning activities [13]. However, current widely used course management systems in university campuses such as Blackboard and WebCT can only offer limited capabilities for adaptive learning experiences. Their support provides relatively weak knowledge representation and reasoning capabilities with a primary focus on instructor-directed conditional sequences [10].

Comparing with the BlackBoard and WebCT systems, the CAPE/eLMS platform emulates human tutors instructing a single pupil on a particular knowledge domain. It provides an opportunity for educators to easily create powerful web-based adaptive learning experiences. Through CAPE/eLMS platform, the educators can assess the strengths and weaknesses of each student and provide adapting learning activities in response. The CAPE/eLMS employs explicit representations of learner knowledge and the knowledge of an expert in the domain. Questioning or interlocution is used to build and revise a model of a particular learner's knowledge and the system is concerned with incrementally aligning this knowledge with that of the expert concerned through the engagement of particular learning content and activities.

Through the CAPE/eLMS platform, the adaptive in-class instruction will be enabled. The classroom activities can be selected from a pre-planned set of alternatives according to the formative assessments. The platform will also enable the interaction between the in-class activities and the adaptive learning activities outside the classroom, such as the preparation for in-class activities and the following up. The interaction can be achieved through dynamic selection of the activities according to the classroom feedback.

The CAPE/eLMS platform can also provide more immediate diagnostic feedback than traditional graded homework assignments and the BlackBoard or WebCT systems. The immediate feedbacks will not only help learners to contemporaneously reflect on their learning, but also contribute to reflection by educators on the overarching learning design. It has been founded by Brosvic et al. that when confronted with previously encountered quiz questions on the final examination, a significant improvement in retention will be achieved if the students were initially provided with immediate feedback rather than delayed feedback or no feedback, and even greater retention when provided with multiple attempts on the initial encounter.

To make it easier to use CAPE-authored learning experience, the eLMS platform can be transparently embedded into BlackBoard and WebCT systems. Therefore, instructors can maintain the rosters and assign eLMS courseware to their learners just as any other kind of Blackboard or WebCT assignment. The instructors can also update Blackboard gradebooks by pulling information from eLMS delivery records.

Undergraduate embedded system design course materials

In this section, the course description, covered topics and laboratories of the undergraduate embedded system design course are described. This course is designed as a technical elective for senior students. It is a 3-credit, semester long course consisting of classroom lectures and laboratory assignments. The classroom lectures are spent on necessary theoretical knowledge. Laboratory assignments are used for students to gain a thorough understanding of the embedded system development phases and familiarity with hardware and software development and debugging tools through hands-on practices.

Considering that software and hardware in embedded systems are interwoven on all levels of design flow from requirements to product specification, and to manufacturing, this course is designed to introduce a balanced view of software and hardware concepts in the design process. The objectives of this course are to introduce concepts, analytical tools and design techniques of embedded systems; to let students investigate topics from hardware/software partition, development tools selection, circuit and operating system design, to system verification and implementation; to let student explore analysis and optimization processes in support of algorithm and architecture design; to let student gain design experience with case studies using contemporary high-level methods and tools; to let student integrate knowledge gained in multiple other core engineering classes into a real-world design; and to let students get experience that is beneficial for securing a job in embedded system design through a design manner similar to that practiced in industry.

The selected textbook is *Embedded System Design*, 2nd Edition, by Peter Marwedel. It provides an overview of embedded system design and relates the most important topics in embedded system design to each other.

In order to adequately prepare the students for the topics of embedded system design, the following courses are required prerequisites to this undergraduate embedded system design course: design of digital systems and laboratory, digital computer structures and engineering visual C++ programming. The knowledge in digital system design is needed for embedded hardware design. The topics should include combinational and sequential logic circuit design, number representation and hardware description language. The software development environment employed in the course uses the C language and the Microsoft Visual C++ compiler, so students must be prepared to program in C++. Students must also understand computer structure and how it affects the execution of their software.

In order to introduce all levels of design flow for embedded system design, topics emphasized in the course include embedded system characteristics; system specification and design techniques; embedded system hardware design; design and implementation of real-time programs and their execution under the control of a real-time operating system; embedded system implementation and validation. Figure 2 shows the content areas covered in the undergraduate embedded system design course.

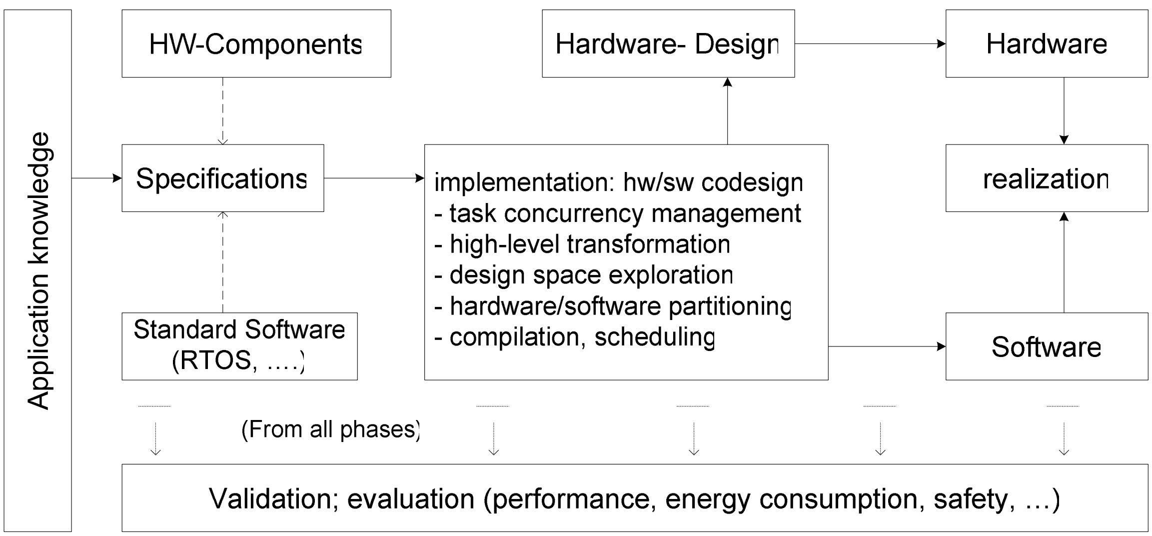

Figure 2. Content areas covered in the undergraduate embedded system design course

During the introduction to the course, students are exposed to basic terminology and concepts. Emphasis is placed on characteristics of embedded system and the challenges of embedded system design. Then emphasis is shifted to a review of requirements for specifications of embedded systems. This is immediately followed by a summary of design techniques for hardware and software portion of the embedded system design and levels of hardware modeling. Since embedded system is information processing system embedded into enclosing products, a hardware design emphasizing those aspects that directly affect the performance of the system is presented. The topics include input/output devices, communications between hardware devices, processing units and memories. Since real-time constraints are the common characteristics of embedded system, embedded operating systems, middleware and scheduling, task communications, time functions are introduced to cover the design and implementation of real-time programs and their execution under the control of a real-time operating system. Once students are familiar with embedded system hardware and real-time programming, hardware/software co-design are included. The overall goal of this topic is to find the right combination of hardware and software resulting in the most efficient product meeting the specification. The final lecture topic is system validation that is extremely important for safety-critical embedded systems. A brief overview over the key techniques for validation and verification are discussed.

To let students gain hands-on experience on "real world" embedded system design and be familiar with hardware and software development and debugging tools, ten required labs utilizing equipment and software available in the departmental labs are assigned to students. These labs are divided into two modules to increase the flexibility of the course materials, one is hardware related labs and the other is software related labs. The educators may adopt one or both in their course according to their particular needs. The hardware-related labs include Introduction to lab equipments; Field Programmable Gate Array (FPGA) design using VHDL; Decoder design; Counter design; Interface keypad with LCD; and Serial port communications. The software-related labs include Introduction to MicroC/OS; Semaphores; Message queues and mailboxes; and Timers and events. Each lab consists of pre-lab work, in-lab experiment, lab analysis and writing lab report.

The details of the covered topics, lecture notes, homework and required labs are available at http://fountain.isis.vanderbilt.edu/teaching/esd. All the students who officially enroll can access these course materials.

CAPE models for the undergraduate embedded system design course

CAPE and eLMS were integrated to provide students adaptive learning experience of embedded system design. This section describes the CAPE models in its current state of evolution. The CAPE models define the design specifications for learning experiences that can be uploaded to an eLMS learning platform for later delivery to learners. The CAPE models introduced in this section offer a generalized framework and can be easily modified by educators to fit their needs.

Courseware models are essentially plans for how learning materials are delivered to learners. The principal content element in a CAPE model is granule. A granule is itself a model that specifies what resources are used to present the granule's content to the learner. A granule's resource model is used to inform CAPE about the required resources and how they are referenced by the content. A granule's resources contain the web-deliverable contents: HyperText Markup Language (HTML) pages, Microsoft Word documentations, or Powerpoint presentations. Therefore, before creating the CAPE model, all web-deliverable course materials must be developed.

Considering that CAPE is a visual language and the courseware model for the undergraduate embedded system design course is very large, four levels are used in the CAPE models to manage the visual complexity, both intellectually and in terms of editing. The top level of the courseware model includes two granules that delivery the welcome information and course introduction through two HTML pages and two phases that contain the lower level models. One phase, Course Overview, is to provide students with the general information of the course, such as course objectives, structure, prerequisites, competencies, and acknowledge, so that they will have an overall vision of this course. The other phase, Modules, contains all the course materials, including lecture notes, homework and quiz. A condition element and an action element are used to provide adaptive sequencing of the delivery of the two phases. If this is the first time for the learner to use this courseware, the condition element will automatically direct him/her to the course overview phase. If the learner has read the course overview, the condition element will let him/her to select whether he/she would like to review the course overview again or go to the Modules phase. Figure 3 shows the top-level CAPE model of the course.

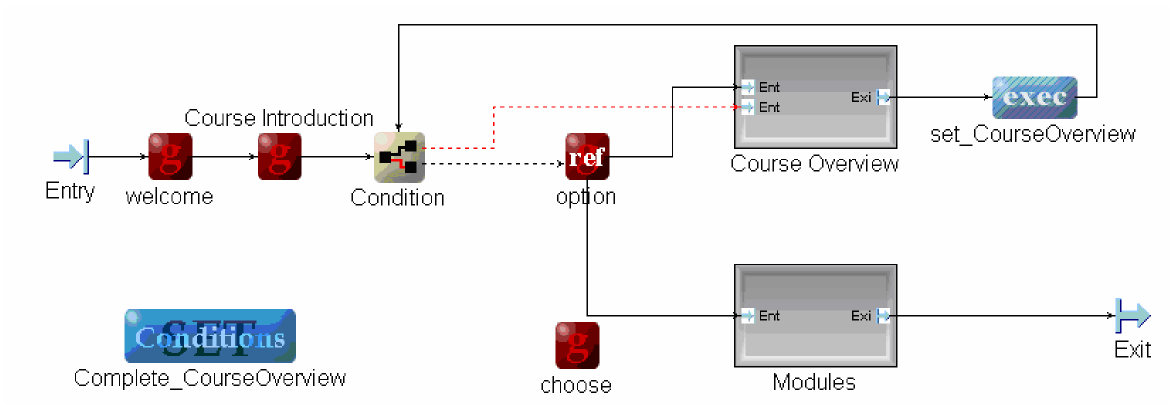

Figure 3. Top-level CAPE model of the course

One of the second-level CAPE model, the Course Overview phase, uses the sequencing to organize the granules in the order of course objectives, structure, prerequisites, competencies and acknowledge. There is no more lower level in this phase. Figure 4 gives the CAPE design of the Course Overview phase.

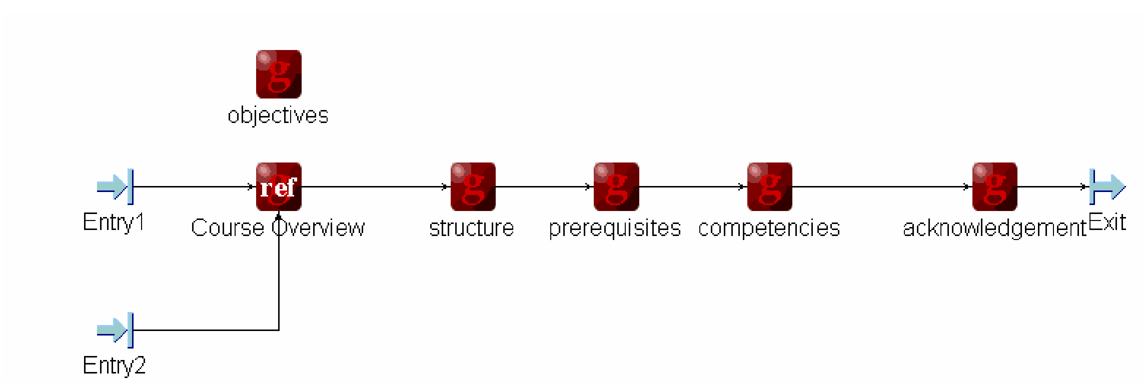

Figure 4. CAPE design of the Course Overview phase.

The other second-level CAPE model, the Modules phase, contains six phases. Each phase represents one major course topic of this course: introduction, specifications, embedded system hardware, embedded operating systems, hardware/software co-design and validations. This CAPE model provides high flexibility of the course contents for other educators. By changing the connection of these modules, the educators may divide the whole course materials into two-or three- semester courses or skip one or two modules to fit their needs. In our case, all the modules are needed. First, all students have to complete module I and II, i.e., introduction and specifications. After successful completion of the first two modules, student can proceed to study hardware design or software design part of embedded system according to their choice. Module V and VI, that is, the hardware/software co-design and validations can be taken after completion of previous four modules. Figure 5 presents the CAPE design of the Modules phase. The condition element and action elements are used determine whether the first four modules have been successfully completed.

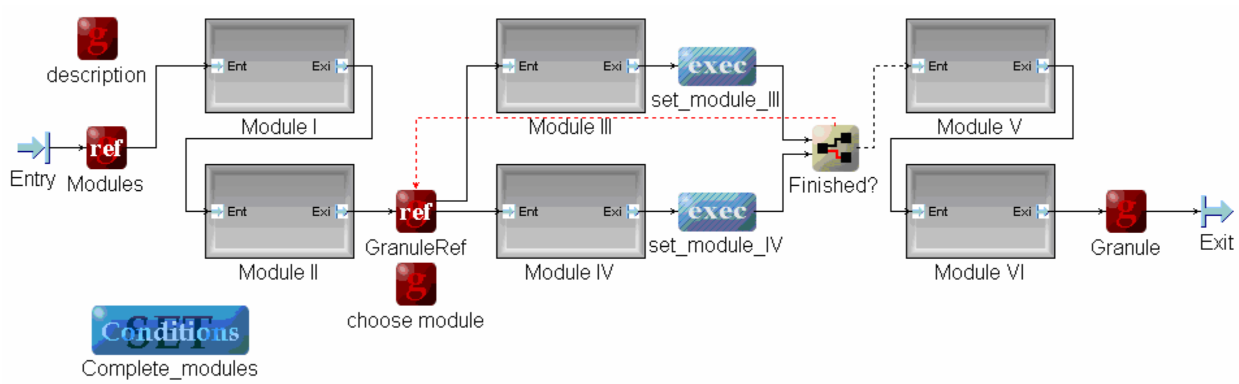

Figure 5. The CAPE design of the Modules phase

The third level of the CAPE model, that is, inside each module, consists of a granule and a phase element. The granule is used to contain the course materials of each major topic of the embedded system design, including the lecture notes in the form of Powerpoint presentation and homework in the form of Word documentations. The phase is used to contain the quizzes that evaluate the students' performance. Figure 6 uses the Module I model as an example of the third level of the CAPE model. The CAPE designs of the other module phase in Figure 5 are similar to Figure 6.

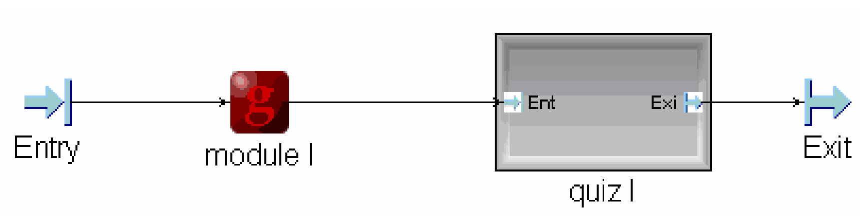

Figure 6. The CAPE model of the Module I

The fourth-level of the CAPE model, that is, inside the quiz phase of each module, is used to determine whether the students successfully complete the major topic. It provides an opportunity for educators to assess the strengths and weaknesses of each student and provide adapting learning activities in response. In each module, five questions including two fill-in-blank questions, two multiple-choice questions and a selection question are posed in the assessment element for performance evaluation. Other educators may specify different questions, or different question types, or different passing requirements according to their needs. CAPE also allows educator to specify different questions to different students to increase the adaptiveness of the courseware. Figure 7 shows the CAPE design of the Module I's quiz phase. The CAPE designs of the quiz phase in the other modules are similar to Figure 7.

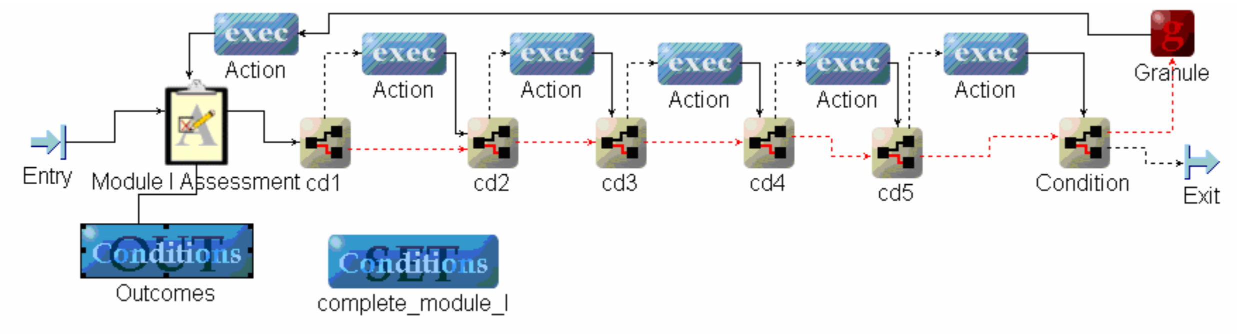

Figure 7. The CAPE design of the Module I's quiz phase

To provide a better view of the CAPE assessment used in our undergraduate education on embedded system design, Figure 8 presents the window shot of the question set for module I assessment as an example.

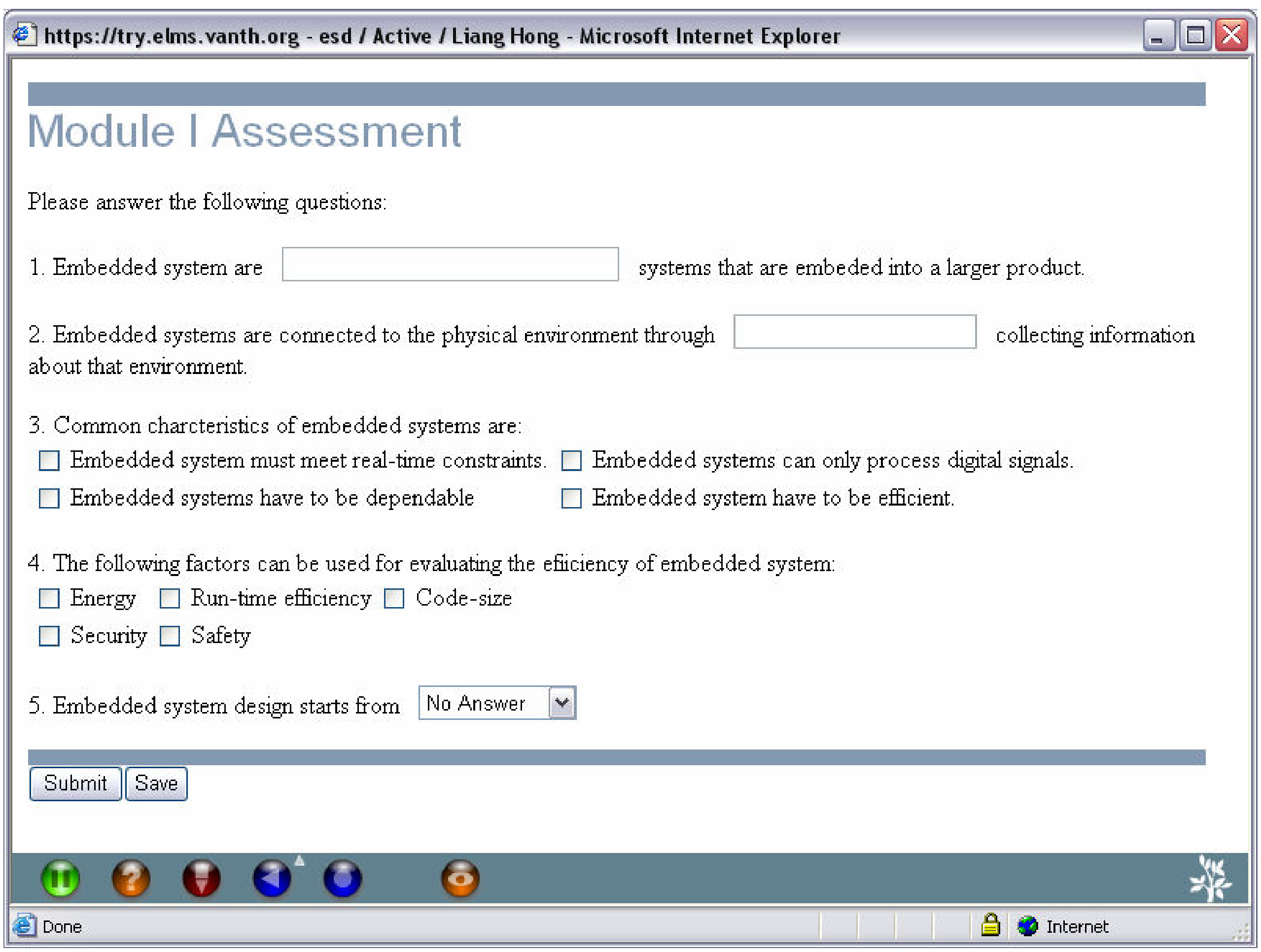

Figure 8. Question set for Module I assessment.

After the student submits his/her answers, a summary showing the score and whether the answer is correct or not will be give immediately. The student can proceed to the next major topic if he/she correctly answers all the questions. If the student does not correctly answer all the questions, he/she will be asked to repeat the quiz. Figure 9 shows the window shot of the summary. This design gives the learners not only the immediate results of assessment but also the opportunity of multiple attempts to significantly improve their retention, motivation and achievement.

The whole courseware through the eLMS can be accessed at http://repo.vanth.org/QuotaLinks/ETjZh.

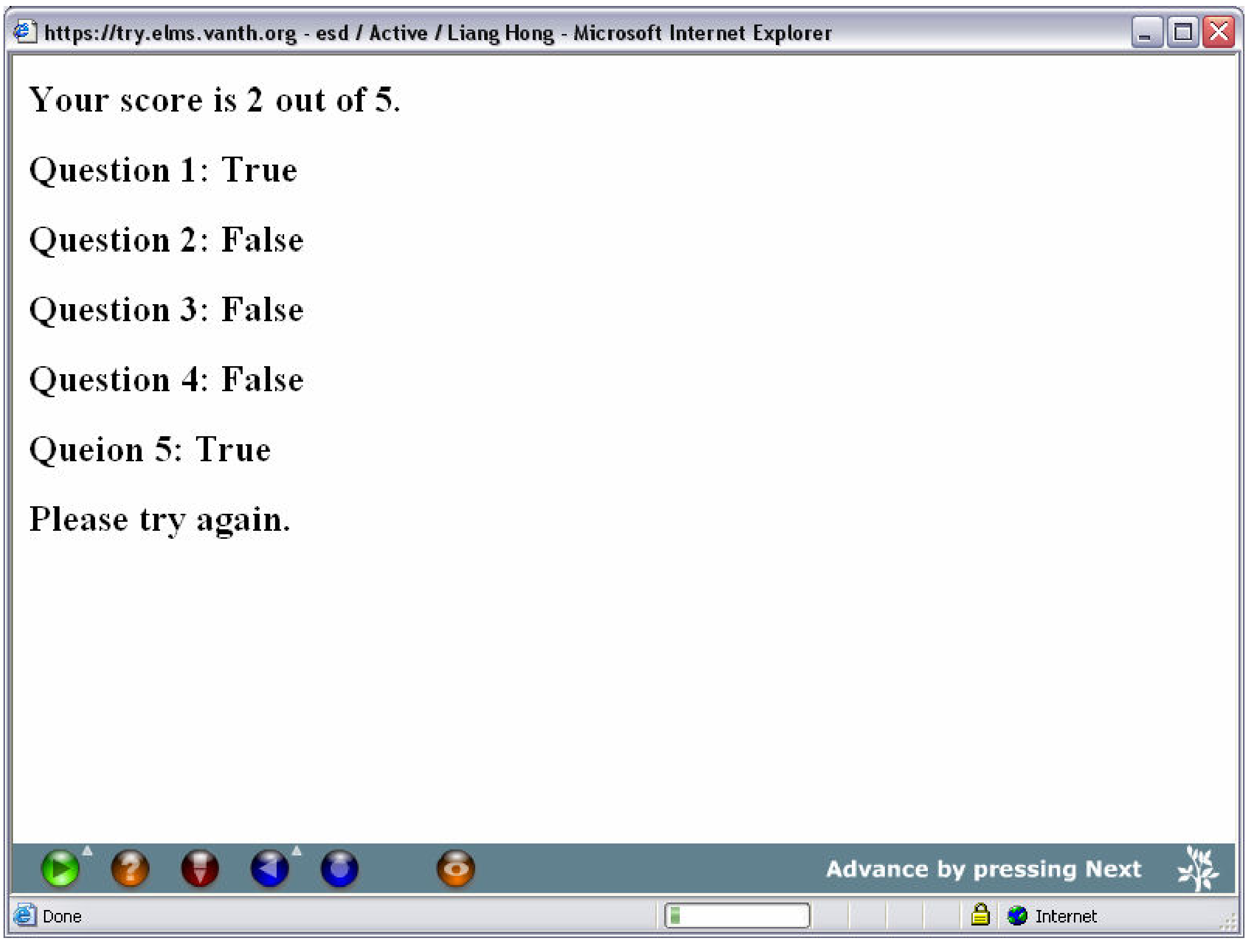

Figure 9. Summary of the assessment results

Conclusions

This paper presents an embedded system design courseware for undergraduate students. It provides students an overview of all levels of design flow from requirements to product specification, and to manufacturing. The course contents, organization of the materials and the required laboratory are illustrated. The CAPE models are built to provide a student-centered adaptive learning environment that addresses learners as individuals, assessing their strengths and weaknesses and adapting learning activities in response. Delivered to learners through the eLMS platform, the CAPE courseware provides efficient education through web-based tools and immediate feedback to the students on their performance. Comparing with other course management software, such as WebCT and BlackBoard, this web-based adaptive learning environment will significantly improve the retention of the learners. It will also provide adaptive interactions between the in-class instruction and the outside classroom activities. The CAPE models introduced in this paper offer a generalized framework. The models can be easily modified by educators to fit their needs.

Acknowledgements

This work is supported by a grant from the Summer Internship Program in Hybrid and Embedded Research (SIPHER). SIPHER is a part of the Information Technology Research (ITR) of National Science Foundation (NSF) funded project, NSF/ITR: Foundations of Hybrid and Embedded Software Systems (Award Number: CCR-0225610). The authors would like to thank Dr. Gabor Karsai at Vanderbilt University for his valuable suggestions and help on the design of the courseware. The authors would also like to thank Larry Howard for his help on the CAPE/eLMS platform.

References

1. P. Marwedel, "Embedded System Design," 2nd Edition, Kluwer Academic Publishers, New York, NY, 2006.
2. G. Borriello and R. Want, "Embedded computation meets the World-Wide-Web," in *Communications of the ACM*, Vol. 43, No. 5, pp. 59-66, May 2000.
3. C. Pahl, "Managing evolution and change in Web-based teaching and learning environments," *Comput. Educ.*, vol. 40, pp. 99–114, 2003.
4. J. A. Zambreno, "CprE 588 – Embedded computer systems," http://class.ee.iastate.edu/cpre588/.
5. L. McClure, "ECEN 4613/5613 Embedded system design," http://ece.colorado.edu/~mcclurel/.
6. Brosvic, G..M., Epstein, M..L., Cook, M..J. & Dihoff, R. E. "Efficacy of error for the correction of initially-incorrect assumptions and of feedback for the affirmation of correct responding: learning in the classroom". *Psychological Record*, Vol. 55 Number 3, 401-418.
7. Bransford J.D., Brown A.L., and Cocking R.R., Editors (1999). How People Learn: Brain, Mind, Experience, and School. *National Academy Press*, Washington, D.C.
8. J. Hartley and D. Sleeman, "Towards more intelligent teaching systems," *Int. J. Man-Mach. Stud.,* vol. 5, no. 2, pp. 215-236, 1973.
9. Harris, T.R., Bransford, J.D. and Brophy, S.P. "Roles for Learning Sciences and Learning Technologies in Biomedical Engineering Education: A Review of Recent Advances". *Annual Review of Biomedical Engineering* 4: 29-48, 2002.
10. L. Howard, Z. Remenyi and G. Pap, "Adaptive blended learning environments," in Proc. the 9th International Conference on Engineering Education, San Juan, Puerto Rico, July 2006, pp. T3K11 – T3K16.
11. http://www.isis.vanderbilt.edu/Projects/VaNTH/
12. P. Marwedel, Embedded System Design, second edition, Springer, 2006.
13. T. Murray, "Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art", *International Journal of Artificial Intelligence in Education*, Vol. 10, pp. 98-129, 1999.