Incorporating Large-Scale Projects into a Multi-Disciplinary Approach to Embedded Systems*

Diane T. Rover[%], Betty Cheng[#], Chin-Long Wey[%], and Matt W. Mutka[#]

[%]*Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA*
e-mail: {wey,rover}@egr.msu.edu; URL: http://www.egr.msu.edu/{~wey,~rover}
[#]*Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA*
e-mail: {chengb,mutka}@cse.msu.edu; URL: http://www.cse.msu.edu/{~chengb,~mutka}

Abstract: Embedded computer systems play an increasingly important role in today's society. In order to adequately prepare today's computer science, computer engineering, and electrical engineering students for their future careers, the special problems with embedded systems development must be adequately addressed in their education. VESL (Visions for Embedded Systems Laboratories) is a project that is designed to address the needs in undergraduate computer science and engineering education. The project develops a multi-pronged approach to bringing embedded systems into undergraduate education. The approach comprised modular course pack development, suitable for alternative teaching models, such as team teaching and development of multi-disciplinary course; team projects to give students hand-on experience with embedded systems; and incorporation of innovative teaching techniques designed to facilitate and enhance the student's learning experience. The multi-pronged approach has been applied and specifically incorporated embedded systems into a suite of relevant courses: software engineering, operating systems, digital system design, and computer system design. This paper reports the concept and practice of introducing objective-oriented modeling to embedded systems development and the application of digital systems design to embedded systems. Students have enthusiastically embraced the courses, citing both the teaming and design projects as being representative of real-word industry experience.

1. Introduction

Embedded computer systems play an increasingly important role in today's society. Such diverse technologies as avionics, automotive drive trains, communication systems, and medical equipment are relying on computers to control system parameters. Although embedded computers are powerful and flexible tools for industry, these very advantages have contributed to a corresponding increase in system complexity. In order to adequately prepare today's computer science, computer engineering, and electrical engineering students for their future careers, the special problems with embedded systems development must be adequately addressed in their education.

VESL (Visions for Embedded Systems Laboratories) is a project that is designed to address three needs in undergraduate computer science and engineering education [1]: (1) providing students with hands-on, real-world experiences in embedded computer system development in an active, collaborative learning environment; (2) increasing accessibility of specialized laboratories to students to allow different learning paces and approaches and to support different contexts and types of interaction; and (3) sharing costly laboratory resources with other universities. To address these needs, we have initiated a three-pronged strategy: (1) instructional module development, (2) laboratory development, and (3) design project development. The strategy has been specifically incorporated embedded systems into a suite of relevant courses: software engineering, operating systems, digital system design, and computer system design.

The curriculum development activities of the VESL project are intended to improve both the instruction and practice of computer science and engineering. Embedded systems are inherently multidisciplinary, meeting the computing needs of scientists and engineers in many domains. Realistic design experiences provide the student with greater insight about the functionality, reliability, and other properties of a system, allowing the system to better serve its domain-specific requirements. In this paper, we report the incorporation of large-scale projects into a multi-disciplinary approach to embedded systems development.

In the next section, we review the VESL project with respect to its goals and objectives. Section 3 describes the concept and practice of introducing objective-oriented modeling to embedded systems development and the application of digital systems design to embedded systems. Finally, a concluding remark is given in Section 4.

* This work is sponsored in part by National Science Foundation grants CDA-9700732, CCR-9633391 and CCR-9901017.

2. VESL Project Overview

The VESL project is led and administrated by the authors and being performed at Michigan State University (MSU) in the context of a newly revised undergraduate Computer Engineering Program within the Departments of Electrical and Computer Engineering and Computer Science and Engineering and in collaboration with three smaller universities in the State of Michigan. They are Lake Superior State University, Grand Valley State University, and Saginaw Valley State University. These universities represent several types of engineering programs (from an electrical/computer engineering technology program to an electrical engineering program) which include an embedded system design component. Their role is to broaden MSU's vision in developing general lab modules appropriate for other institutions' use; to assist in lab module development to meet any specific needs of the participating institutions; to provide consultation on educational innovations that impact the institutions; to provide partners to investigate and test the networked operation of the lab and evaluate its capabilities and limitations; to incorporate VESL instructional modules into respective curricula as appropriate; and to be representatives of the VESL project.

The VESL name highlights an important aspect of the curriculum development: use of laboratories to support integration of material throughout a student's program. Moreover, the name is symbolic of the connective and open features of the developed laboratory. It is intended to support connective learning, facilitating closer interaction among students and instructor. It is designed to connect students with real systems and real problems. The course materials resulting from this project are accessible to students on and off campus via the Internet, so that 'open' takes on an extended meaning encompassing accessibility, networked/interoperable systems, distance education, and resource sharing.

The focus for the curriculum development is embedded computing systems. Research results in real-time computing, software engineering, digital system design, and wireless systems are being transferred to the instructional domain. We are applying a concept-driven, modular development approach that is called '*Open Your I's*' (to active learning). The premise of the approach is that concepts are learned by a process of *Introduction*, *Instruction*, *Illustration*, *Investigation*, and *Implementation*. For example, concept illustration refers to live or multimedia demonstrations in the classroom; concept investigation refers to hands-on or multimedia interactive exercises; and concept implementation refers to hands-on laboratory projects. The development of 'I' modules facilitate integration across the curriculum, including cross-pollination of subject matter, as well as dissemination to other universities. Additionally, it provides a structured means to transfer techniques and tools from research into instruction.

3. Course Module Development and Team Projects

The VESL project develops and implements a multi-pronged approach to curriculum development that specifically incorporates embedded systems into a suite of relevant courses. Special emphasis was placed on incorporating recent results from embedded systems research. We start at a high-level by describing the software engineering course that addresses how embedded systems are defined and requirements are analyzed. Next, we describe how the operating systems course presents system software and services relevant to embedded systems, particularly real-time operating system concepts. The emphasis is placed on the development of real-time and mobile computing systems. Finally, we describe the digital system design courses that investigate how components of embedded systems are designed, implemented, integrated, and tested, ranging from microprocessors, to integrated circuits, to interfaces. We focus on providing bridge to hardware environment and the issues of hardware-software co-design and implementation.

This section describes the concept and practice of introducing object-oriented modeling to embedded systems development and the application of digital system design to embedded systems development.

3.1 Introducing Object-oriented Modeling to Embedded Systems Development

The Software Engineering (SE) course is designed to teach students the fundamentals of software development, beginning with problem identification and requirements analysis through design and testing. Significant emphasis is placed on object-oriented analysis and design techniques.

Software engineering education has been largely directed towards mainstream software-based applications, such as the development of client-server information systems, network management systems, and CASE (Computer-Aided Software Engineering) tools. Most of the techniques taught in the software engineering courses are generically applicable to software-based systems, with little emphasis placed on hardware environments. In order to

complement the current material presented in the software engineering courses and to provide training in an area that is gaining increasing attention, we extended the SE course to cover software development techniques specific to embedded systems in addition to those that are generally applicable to software systems.

The course has had a focus on object-oriented development techniques. But we have found that, traditionally, embedded systems have not been developed within the object-oriented paradigm. And even within the structured analysis and design paradigm, there seemed to be a lack of emphasis on a stepwise refinement process. One of our current research projects is to explore how a commonly used object-oriented modeling technique (UML [2]) can be used to model embedded systems [3] and how the models can be used as the basis of formal specifications [4]. The formal specifications enable developers to perform numerous analysis tasks, such as checks for completeness and consistency of the diagrams, which would otherwise only be analyzable by visual inspection. Therefore, the formalization of UML bridges the gap between the graphical, easy to use notation that might be ambiguous and prone to errors with formal specifications that enable rigorous analysis and verification activities.

In order to use UML to model embedded systems, we had to introduce new guidelines for model development. UML comprises more than eleven diagrams, but we found that four diagrams were sufficient to model the structure and the behavior of embedded systems. First, the use case diagram describes goals or objectives of the system and depicts the actors of the system. They can also be considered as a representation of a collection of scenarios of system use. Second, the object model describes the static, structural aspects of the system, that is, the objects and the relationships between the objects; this model is similar, in form, to the entity-relation diagrams traditionally used to model database systems. Third, the state diagram is used to describe states of the system and the events that cause the system to transition between the states; the state diagram is used for the dynamic model. Finally, the sequence diagram capture a specific scenario of the system. That is, a sequence diagram is a specific path of states and transitions through the state diagram. We found that for all embedded systems, the use case diagram, the object model, and the sequence diagrams were fairly straightforward to develop. In contrast, the state diagram for an embedded system can become quite complex given the potentially large number of combinations of inputs from sensors and the number of possible responses to be effected by the actuators. The students also had to model the concurrency that is almost always exhibited by the behavior of embedded systems. The sequence diagrams were used to "test" or validate the behavior of the system depicted by the state diagram.

In order to further enhance the students' experience with the SE course, we solicited embedded systems projects from industry. The students worked in teams on these projects, where each student turned in weekly status reports and each group met at least once per week. Each team had a project manager, a documentation manager, a facilitator (who scheduled meetings), and a research coordinator (who was in charge of collecting domain information). This year, the projects were from the automotive domain, including a smart cruise control system, an integrated starter, and diesel engine controller. The students had a customer with whom to ask domain questions and clarify project specifications. Having projects from industry helped students to understand the context for class and laboratory materials. This extra dimension to the course also motivated the students in their project work given that there was a customer awaiting the results of their projects. While the students only completed the requirements analysis, design document, and prototype deliverables, the experiences gained has been invaluable even without a full-blown implementation. The students felt that the experienced from this course was instrumental in their successfully completing the capstone course the following semester. The customer has indicated that the project documentation has helped their organization to learn how object-oriented technology can be applied to embedded systems development. They were particularly interested in the model checking component since they had previously considered that technique to be too difficult or complex to adopt in their organization.

In addition, students were taught how to identify safety-critical aspects of the system and how to use the formal specification language, Promela, to formally specify these critical conditions. Promela specifications could then be executed in order to simulate the behavior of the model, or we could prove properties about the Promela specification using a Model Checking tool called SPIN [5] developed by Lucent Technologies. We define the term `safety-critical' conditions to refer to those constraints that if violated may cause something `bad' to happen, that is, loss of life, property, or money. This exercise greatly helped to demonstrate to the students the value of the process of formally specifying critical conditions. Industrial specifications, particularly from the automotive domain, made it easier to understand why it is necessary to be able to prove safety critical properties. Students found that English descriptions could be ambiguous and incomplete, and sometimes even contradictory. By going through the process of representing the conditions in terms of a formal language, students detected these inconsistencies and ambiguities, even without tool support. These findings are in line with what our experience has been with industrial collaborators. Then SPIN was used to simulate the behavior of the state diagrams as capture by the Promela specifications. The students were able to encode the sequence diagrams in terms of Promela specifications and verify that these

scenarios were represented in the state diagrams. The students also verified safety properties and liveness properties using SPIN.

As another means to emphasize the importance of careful software engineering practices, we organized peer reviews (also known as document inspections) of one of the deliverables. Each review team consisted of the project manager for the project, a scribe from the team, two external reviewers, and a facilitator (the instructor or a teaching assistant). The external reviewers were given one week to carefully review the document, looking for syntactical and content errors. The project manager prepared a brief oral overview of the project, and the scribe was responsible for recording the comments during the review. The students found this exercise to be extremely useful and enlightening. The reviewers found that they learned better how to write their own documents from reviewing someone else's document. The project manager and scribe learned how to better present their ideas and to accept constructive criticism.

3.2 Application of Digital System Design to Embedded Systems

Three undergraduate courses courses represent the extent to which the embedded system theme supports the curriculum in digital system design. In the introductory Digital Logic Fundamentals course, students are motivated to appreciate the role of digital logic in their daily life through a group assignment to identify a product or process driven by digital logic (e.g., traffic lights, digital camera, calculator, microwave oven, etc.). They select, investigate, and discuss a product and write a web-based, electronic essay. Thus, while students are learning the basics, they begin to see the relevance in something tangible. It raises their awareness about computer-based systems as well as raises their curiosity. Heightened student interest improves learning of the fundamentals and provides a basis for subsequent courses in digital system design. In the senior-level Digital Electronics course, students learn in-depth about digital circuits, VLSI design, and application-specific integrated circuits (ASICs). Students are instructed that ASICs are an important hardware component of embedded systems. Students are informed of related topics such as hardware-software codesign, systems-on-a-chip, and IP (intellectual property) cores. Student design groups must address these topics in their projects. The capstone course, Computer System Design [6], is the major engineering design experience for seniors in computer engineering. The major engineering design experience involving embedded systems to control process and contemporary hardware/software design tools and practices. In addition, two graduate courses, advanced VLSI design and Embedded System Design, were also developed. The latter course [7] includes the following topics: Hardware/software system and codesign; Modeling of computations for embedded systems; Modeling, specification, synthesis, and verification; Hardware/software implementation; Performance analysis and optimization; Design methodologies and tools.

Students learn about embedded systems, i.e., electrical systems that contain embedded computers to control processes. At the completion of this course, each student has actively participated as a member of an engineering design team and made significant contributions to achieving the team's stated goal and objectives. Student projects have centered on use of tools and technologies such as LabVIEW, in-system programmable logic, 68HC11 microcontrollers, StateCharts, etc. In addition, projects have involved the use of the web for remote monitoring and control of the embedded system, thus broadening the scope to distributed systems and requiring more advanced development strategies such as hybrid prototyping. A number of 'fun' design projects have been developed in the capstone course offered recently [6]: Electronic Acceleration Monitor, Soccer Kick Trainer, Thermoelectric Mobile Refrigerator, Audio Signal Processor, Optical Display, Health Care Information Appliance, the Lisbon Endeavor, and etc. Among the projects, the Lisbon Endeavor is a sponsored project. Six students formed a team to design, build, and test a fully collaboration system using the PC/104 wearable computer. The system meets the design specification given by the sponsor. The system enables two remote users to establish communication with each other over a wireless LAN (Local Area Network). A user-friendly GUI interface was built to allow a user to operate a RC car over the web by using the wearable computer and wireless LAN.

In the capstone course, a 'cross-functional teaming' approach is applied. Students are grouped into two sets of interdependent teams, 'design teams' and 'skill teams.' Design teams are formed for the entire semester. Each of these teams works on a specific engineering design project that involves the collaborative development and evaluation of a 'product' that contains an embedded computer. Skill teams are formed from representatives of each design team. As the name implies, these teams 'learn' specific skills needed to ensure success within the individual design projects. Skill teams are highly focused, and the intent is to foster self-directed learning in the interests of lifelong-learning as well as learning by teaching others (since skills brought back to design teams must be shared with other members). Teams were put into a context of a single company's engineering staff meeting a customer's needs. The company, Spartan Embedded Technologies, issued a request-for-proposals (RFP) and the design teams submitted written proposals and gave presentations. During the term, teams spent considerable time on written and

oral communication, including progress reports, technical reports, final reports and demonstrations, and websites. Students gave all presentations using PowerPoint, a projection system, and a notebook computer in the classroom. Design projects contained a number of common threads that facilitated project success through the use of skill teams and cross-functional teaming. Students were introduced to strategies for effective teaming, group processing, and self-assessment, including the periodic use of evaluation forms for the team leader, each member, and the team as a whole. The cross-functional teaming model is intended to support multidisciplinary teams. We have thus far involved only faculty and student facilitators from mechanical engineering in the design projects, as logistics are not yet in place to build teams including students from other disciplines. Embedded systems have provided a natural context for multidisciplinary projects.

Students have enthusiastically embraced the course, citing both the teaming and design projects as being representative of real-world industry experiences. Feedback has been obtained through course evaluations, journals, and assessment reports. Individual students maintained a journal, in which they entered impressions of the course and identified how they were fulfilling the learning objectives. At the end of the term, each student wrote a 'professional self-assessment report.' In this report, they assessed their learning in the course, the impact of this course, and their career plans. Student feedback has emphasized confidence in becoming an engineer; career planning and interviewing for jobs; understanding the wisdom of teaming; oral and written communication skills; industrial needs, requirements, and practices; and lifelong learning. Students enjoyed tackling open-ended design problems and were extremely satisfied, albeit surprised in some cases, with the extent of their learning under this course model. Students often remarked that this was one of the most time-consuming and challenging courses ever taken and, at the same time, was one of the most exciting and rewarding.

4. Conclusion

The curriculum development activities of the VESL project are intended to improve both the instruction and practice of computer science and engineering. Embedded systems are inherently multidisciplinary, meeting the computing needs of scientists and engineers in many domains. Realistic design experiences provide the student with greater insight about the functionality, reliability, and other properties of a system. In this development, several new dimensions were added to help students understand the importance of good software engineering practices for embedded systems. These changes were motivated by the fact that embedded systems are typically 10 to 100 times more common than their desktop counterparts [9], residing in everything from engine systems, to toasters, to autopilots. The initial feedback from the students and our customer has been extremely positive. There has been a true implementation of technology exchange, not just technology transfer. It is noted that the new techniques that we have introduced to this course can also be applied to other domains. On the other hand, students have enthusiastically embraced the computer system design course, citing both the teaming and design projects as being representative of real-world industry experiences. Employers and external advisory boards have provided positive feedback. For example, on the subject of student frustrations with time demands of reporting and teaming, one employer said that if a student isn't spending 90% of his/her time writing or communicating with others, he/she is not communicating enough. Others have commented that the teaming and open-ended design experiences are very valuable and realistic. In addition, employers have found the course website to be informative and useful.

References

1. M. Mutka, D.T. Rover, C.L. Wey, and B.H.C. Cheng, "Visions for embedded systems laboratories" Michigan State University, Web. NSF Combined Research-Curriculum Development Program, URL: http://www.egr.msu.edu/VESL.
2. J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*, Addison Wesley, 1999.
3. W.E. McUmber and B,H.C. Cheng, "UML-based analysis of embedded systems using a mapping to VHDL," in Proc. of IEEE High Assurance Software Engineering (HASE99), Washington D.C., November 1999.
4. E.Y. Wang and B.H.C. Cheng, "Formalizing and integrating the functional model into object-oriented design," in Proc. of International Conference on Software Engineering and Knowledge Engineering, June 1998. (Nominated for Best Paper)
5. G.J. Holzmann, "The model checker SPIN," IEEE Trans. on Software Engineering, Vol. 23, May 1997.
6. EE 482, Michigan State University, http://www.egr.msu.edu/classes/ee482/.
7. EE 809, Michigan State University, http://www.egr.msu.edu/classes/ee809/
8. D.T. Rover and P.D. Fisher, "Cross-functional teaming in a capstone engineering design course," Proc. of the 1997 IEEE/ASEE Frontiers in Education Conference, November 1997.
9. B.P. Douglass, *Real-time UML*, Addison-Wesley, 1998.