

## **A Case Study of Interdisciplinary Teaching at Kansas State University**

**M. Mizuno<sup>1</sup>, D. Lenhert<sup>2</sup>, M. Neilsen<sup>1</sup>, G. Singh<sup>1</sup>, N. Zhang<sup>3</sup>, and A. Gross<sup>4</sup>**

<sup>1</sup> Department of Computing and Information Sciences, Kansas State University (KSU) { masaaki, neilsen, singh } @cis.ksu.edu\*

<sup>2</sup> Department of Electrical and Computer Engineering, KSU, lenhert@ksu.edu\*

<sup>3</sup> Department of Biological and Agricultural Engineering, KSU, zhangn@ksu.edu\*

<sup>4</sup> The IDEA Center, 211 S. Seth Child Road, Manhattan, Kansas, agross@ksu.edu\*

### **Abstract**

At Kansas State University the Electrical and Computer Engineering department (EECE) and the Computing and Information Science department (CIS) have collaborated in developing with an NSF CRCD grant a series of embedded systems courses. In a minimum of credit hours, the CIS students learn more about the interconnection of hardware and its effect on software decisions and the EECE students learn more about computing theory especially real-time scheduling theory and verification. Either group could do this by taking normal courses but it would require many more credit hours of present day courses. These courses were designed for any engineering student to be able to take.

The paper will present the gains obtained by the faculty and the students taking the courses as evaluated by an independent group. The possible disadvantages of this approach will be discussed, although we did not encounter many of them. Recommendations for other groups interested in developing multi-disciplinary courses are made.

### **1 Introduction**

Traditionally, the Electrical and Computer Engineering department (EECE) and the Computing and Information Science department (CIS) have tried to minimize the overlap of materials taught in their courses. This has resulted in wide gaps in the course material especially in the area of

---

\* This work was supported in part by the National Science Foundation under NSF-CRCD Grant #9980321.

how one area affects the decisions in the other area. CIS departments have traditionally concentrated on theory, concepts, programming in high-level languages on protected OS environments, and architecture from the point of view on how it affects the software. EECE departments have traditionally concentrated on gate-level descriptions of hardware, architecture from the hardware perspective, interconnection of hardware, and low-level device drivers/interrupt handlers in assembly language on primitive development environments.

This separation does not work well in the area of embedded systems where a broad knowledge is required in areas traditionally not covered in any one discipline. Consequently, it is very difficult to train students and engineers within a single discipline to effectively design and implement complex, real-time embedded systems.

This paper will present a brief outline of the interdisciplinary course we developed and a summary of the evaluation of these courses by both the students and faculty as taken by an independent group. Then the benefits and disadvantages of this approach will be presented.

## **2 Courses Developed**

In response we have developed an interdisciplinary set of courses for education in real-time embedded system design.<sup>1</sup> This set of courses consists of four courses with the first two courses consisting of 3 separate modules. The first course was designed so that students from any engineering discipline could take whichever modules were needed to prepare them for the following courses. The remaining three courses were designed so that computer science, and computer engineering students would have the background to start with the second course though possibly skipping one or more modules in it depending on their individual background.

The first course is designed to be a remedial course for students who do not have a proper background for the subsequent courses in the proposed course sequence. The course consists of three modules; students can take only the necessary modules and earn one credit each. The first module is Basic Real-Time Electronics. This module covers basic digital logic, logic families, transistors and how to drive motors and relays from a simplified approach. This module was taught in an existing microprocessor lab housed in the EECE department. The second module is Data Structures. In this module, we intend to cover basic data structures such as stacks, queues, lists, and priority queues, and algorithmic techniques for sorting, searching and hashing. We will also introduce the concept of object-oriented design and interface specifications. The third module is Concurrent Programming. In this module, we present concepts of processes, threads, and synchronization using programs found in traditional computer science textbooks such as the dining philosopher, reader/writer and producer/consumer.

The second or implementation course also consists of three modules. The first module is Real-Time Programming Fundamentals. We present the precise relationship between each C/C++ construct and corresponding assembly code generated by compilers. Then, we introduce special techniques to implementing microcontrollers, such as initialization of programmable CPU modules and peripheral devices, linking technique to produce ROM-able code. The second module is Real-Time Operating

Systems. In this module we present the details of a microkernel, how it is used and which microkernels are typically available for various microprocessors. The third module is Real-Time Embedded Systems. This module deals with interconnection of more complex peripherals than ones presented earlier, such as CAN networks, DA/AD converters, and PWM (Pulse Width Modulation) - -- all of such devices come with variations of various microprocessors. Through such lab projects, students learn how to control various peripherals and build a (small but) complete real time distributed microcontroller system. With the NSF CRCD grant, twenty Phytex Microcontroller boards containing an Infineon C 167 microcontroller and Tasking C/C++ compilers were purchased. These setups were used in each of the modules. The first two modules were taught in the CIS department and the microcontrollers were used in an existing CIS laboratory. The third module was taught in an existing microprocessor laboratory in the EECE department. The microcontrollers and software were then moved to this lab. Additional hardware needed for the small design project was purchased with funds from the contract with some supplement from both departments about equally.

The third course is the theory course. This course is intended to teach techniques for design and analysis of an embedded system. The course directly imports ideas developed as part of our research in embedded system. The aim of this course is to provide students with a strong theoretical foundation in designing and analyzing embedded systems. The content of this course covers real-time scheduling theory, verification of concurrent programs, and elements of the requirements, design and implementation phases.

The fourth course is the capstone design course. This course is intended to teach techniques that allow engineering an embedded system to satisfy certain performance requirements. The students must be able to evaluate various design choices and make decisions accordingly. A major component of this course is an industrial-sized involving the design and implementation of a complete embedded system. This course was taught in the EECE department with assistance from the BAE department using the same laboratory as the second course. Special hardware for this module was purchased under a Motorola grant to this laboratory or borrowed from the BAE department.

Traditionally, many cross-listed courses are taught differently each semester depending on which department is teaching the course. Also, there may be little or no interaction between the faculty teaching cross-listed courses. In contrast, our courses have been developed through continual interaction and revisions to ensure a smooth transition for students. Finally, these courses are electives in all departments, so they do not need to meet specific departmental program outcomes; they are truly interdepartmental.

### **3 Evaluation**

At the completion of each module offered during the Spring 2000, Fall 2000, Spring 2001, and Fall 2001 semesters, students were asked to complete an evaluation developed by an external evaluator with input from course instructors. Seventy different students completed at least one module in the curriculum during the first four semesters. Most were graduate students with six (9%) being upper level undergraduate students. Evaluation survey response rates ranged from 73-100% with an

average response rate of 89%. The student evaluations consisted of 4 major parts. These were course process, amount of prior knowledge, amount of learning, and overall evaluation. Each part was marked on a five-point scale. In addition, comments were requested on suggestions for improving the course and relationship to prior modules.

The faculty involved also filled out an evaluation developed by an external evaluator at the end of the first year and again at the end of the second year. These questions related to possible changes in the modules, strengths and challenges of the curriculum, changes in the process, support from the departments involved, recommendations to other groups, etc. Specific comments will be indicated below.

#### **4 Benefits**

There are considerable benefits for both the students and faculty from this type of approach. The faculty gains from working closely on a common project with faculty from other departments. In this case the computer science faculty gain insights and knowledge in the area of hardware and its interconnection to the software. The electrical engineering faculty gains in the increased knowledge of software and how to use it more effectively. The computer engineering faculty gains in the area of the integrated combination of both hardware and software plus the increased knowledge of the theoretical basis of the software. The students gain by being exposed to the integrated combination of software and hardware. Thus increasing their knowledge of each area, but mainly the combination and how each affects the other.

At the end of the first year, the faculty responded to the question by the independent evaluator about the strengths of the program as summarized in the following manner.

Overwhelmingly, the faculty noted the strengths of an interdisciplinary program because it integrates advances in both science and engineering, making the curriculum very comprehensive. Having students from different disciplines in the same course also add the quality of the experience. Additionally, students can learn the necessary material for embedded systems in a few semesters. The relevance of the content area to industry was also noted as a strength.<sup>2</sup>

After the second year the faculty responded:

AA strength noted by a number of the faculty was the curriculum's emphasis on theory, practice, and the use of contemporary topics. Additionally, the interdisciplinary exposure and increasing student understanding of the similarities and differences in hardware and software design was noted as a strength. The fact that the curriculum is supported by an interdisciplinary faculty with balanced backgrounds in theory and practice adds to the strength of the curriculum.<sup>3</sup>

All of the faculty members involved were overwhelmingly positive about the outcomes being worth the effort.

The students were asked to evaluate the effectiveness of different course elements in terms of how they contributed to their learning. The responses of the students as evaluated by an independent source were uniformly positive, indicating that the material in one module was made clearer by its uses in a following section. The students commented that the combination of modules had a good compliment of hardware and software and that the combination made them more confident that they could design a real-time embedded system. Additionally, the students were asked to evaluate their prior knowledge about specific course content on a five-point scale and to evaluate their learning of specific course objectives on a five-point scale. There was a diverse student response to previous knowledge, but the students were generally quite positive about the learning on specific course objectives.<sup>1</sup>

## **5 Disadvantages**

The major disadvantage that we encountered with our approach was that we underestimated the wide variety of backgrounds and skill of students from disciplines other than Computer Science, Computer Engineering and Electrical Engineering. While we developed three 1 credit segments in our remedial course to put the students from other engineering disciplines on an equal footing, the variation in skills was too large. We would expect industrial participants to have an even wider range of skills. We are looking at a possibly changing the process to a different type of multiple entry system where, for example, those not needing the theory would take a different path than those where this was more important.

The compression of the material in just a few credit hours caused us to remove almost any overlap of material between the courses in the interest of time. This makes the courses very non-redundant. While this seems good from an efficiency point of view, most students (and faculty) need some redundancy or repetition of material in order to understand the framework for the new material. There are several ways of overcoming this, but all will require much more work on the part of the faculty and students. One method is to prepare material (either web based or handout) that summarizes the material that is considered background for this particular segment. As an example, one might consider the preparation of material on the terms that will be used, their definition and how they are used for non-Electrical or Computer Engineering students when discussing motor speed control using pulse width modulation. This material would be made available to the student a day or two before the class that used this information.

A disadvantage that we did encounter on the part of the faculty was the increased time commitment needed to either attend the other classes or making sure that they understood the material and method of presentation for the modules that preceded the module taught by the faculty member. Also there was an increased number and length of meetings needed to keep things organized. While this extra effort on the part of the faculty member could be considered a disadvantage, all the faculty members felt that the benefits outweighed the disadvantages.

There are other possible disadvantages that we did not encounter on this project, but may occur in some universities and departments. The first of these is the concept of “interdisciplinary”. While this is a common buzz word in most departments, when a faculty member is involved with another department, the faculty member’s administration may feel that the department or college is being short-changed as the faculty member is not always visible working on things deemed important for the department. Also this can easily increase a faculty member’s time commitment since the number of meetings they must attend increases and it is impossible to attend half of a meeting. We did not encounter any problem here other than an increase in the number of meetings to attend. The department heads were very agreeable and occasionally met on an informal basis to discuss this project and other common areas. Additionally the departments are in a common college. Another possible problem is that if the department has not been involved in interdisciplinary work, the administrators may have difficulty evaluating the faculty member properly.

## **6 Conclusions and Recommendations.**

The development of the interdisciplinary curriculum aided all the faculty members both in teaching and research and they would do it again.<sup>2,3</sup>

The faculty recommended the following suggestions to groups attempting to implement such an interdisciplinary curriculum:<sup>2,3</sup>

- Create a team that can work well together and be committed to the project. This is critical, as it is easy for a multi-disciplinary team to lose focus.
- Have regular meetings and expect the administrative and time requirements to be substantial.
- Involve students from different departments.
- Develop and review a reasonably complete course outline for each module prior to the first offering.
- Hire an external evaluator.
- Overall, it can be very rewarding to work with engineers in different disciplines.

## **References**

- <sup>1</sup> M.L. Nielsen, D.H. Lenhert, M. Mizuno, G. Singh, N. Zhang, and A.B. Gross, “An Interdisciplinary Curriculum on Real-Time Embedded Systems”, in Proceedings of the 2002 American Society for Engineering Education Annual Conference and Exposition, Montreal, Canada, 2002.
- <sup>2</sup> A.B. Gross, “Year 1: Faculty Survey Summary, NSF-CRCD Grant #9980321”, The IDEA Center, December 4, 2000.
- <sup>3</sup> A.B. Gross, “Year 2: Faculty Survey Summary, NSF-CRCD Grant #9980321”, The IDEA Center, September 27, 2001.

## **Biographical Information**

MASAAKI MIZUNO is a Professor in the Department of Computing and Information Sciences at Kansas State University. His research interests include operating systems and distributed systems.

DONALD H. LENHERT is the Paslay Professor in the Department of Electrical and Computer Engineering at Kansas State University. His research interests include embedded systems and digital testing.

MITCHELL L. NEILSEN is an Assistant Professor in the Department of Computing and Information Sciences at Kansas State University. His research interests include real-time embedded systems, distributed systems, and distributed scientific computing.

GURDIP SINGH is an Associate Professor in the Department of Computing and Information Sciences at Kansas State University. His research interests include network protocols, distributed systems, and verification.

NAIQIAN ZHANG is a Professor in the Department of Biological and Agricultural Engineering. His research interests include sensors and controls for biological and agricultural systems.

AMY B. GROSS, Associate Director of The IDEA Center, served as the external evaluator for the NSF-CRCD grant. The IDEA Center's mission is to assist colleges and universities assess and improve teaching, learning, and administrative performance.