Real-World Engineering: A Course for Masters Students Headed for Industry

Sally A. McKee[1] *and Diane M. Kubarek*[2]

Abstract *— This paper describes an experimental, project-oriented course whose technical content focuses on the interface between hardware and software. The underlying agenda for the course is to teach students — in one semester — as many skills as possible for succeeding in industrial engineering environments. In essence, we are functioning as part of a small company for one semester. The non-technical portion of the course focuses on communication in speaking and writing: within a project team, with rotating "Project Leaders", among separate teams, and with the "CEO" (the first author). The students themselves are helping to create the project part of the course, and the Cornell Academic Technologies Center (via the second author) is supplying tools to enable and evaluate the communication portions of the course. Guest speakers from diverse branches of high-tech industry provide perspective and advice on career paths for graduates with engineering degrees.*

Index Terms — communication skills, hardware/software interface, managing teams, mentoring

INTRODUCTION

The Cornell University Masters of Engineering program offers advanced training in any of thirteen engineering fields, and usually takes two semesters to complete. It serves as a professional extension of a four-year undergraduate engineering program emphasizing basic math and science: the Master of Engineering (MEng) curriculum builds on those skills, but focuses more intently on application. In addition to requiring advanced technical courses in the student's chosen field, the program requires that the student complete (and report on) a design project under the supervision of a faculty member. A few of the graduating MEng students go on to complete doctoral research degrees, but most take industrial positions, often with the intention of later attending law, business, or even medical school.

THE ORIGINAL VISION

The ideas that gave rise to this *very* experimental course (Electrical and Computer Engineering 595: Real-World Engineering) came from three distinct, largely orthogonal goals:

- to help students find fun, challenging design projects,
- to teach students more about the hardware/software interface, and
- to help students build and practice effective communication and leadership skills.

The first goal (the initial inspiration for the course) was intended to help students design inter-related projects that are part of a larger whole, much like the projects they'll be working on in industry. It also attempts to free other faculty members so that they may take on MEng student advisees only when those professors have projects of appropriate scope, matching the students to projects, instead of trying to find projects for arbitrary students.

The second goal arose from the realization that many of the students with Electrical and Computer Engineering undergraduate degrees understand hardware (from gates to microprocessors) quite well, but that their programming skills are limited to Hardware Description Languages or (the ever popular) Java. Few students had taken Operating Systems or Compilers, or had worked on low-level software (e.g., device drivers, or even command interpreters) that interacts most directly with the hardware. Likewise, many of the students coming from a traditional Computer Science background lack hardware lab experience, either building devices or physically experimenting with them to better understand their behavior. Why not bring these groups of students together, give them a common goal, and let them learn from each other? Understanding that blurry line where the hardware and software meet is essential to being a good computer engineer or systems software engineer. The former group of engineers needs to know how their creations will be programmed, and the latter group needs to communicate to the former what the hardware requirements are for programmability.

The third goal arose from conversations with the people in industry who are potential employers of our graduating MEng students. One of the first qualities they look for is effective communication skills, both in speaking and in writing. Since most technical courses focus on building technical skills, a class that also emphasizes communication seems interesting and appropriate.

[1]Sally A. McKee, Cornell University, School of Electrical and Computer Engineering, 324 Rhodes Hall, Ithaca, NY 14853, sam224@cornell.edu
[2]Diane M. Kubarek, Cornell University, Academic Technology Center, 126 Computing & Communications Center, Ithaca, NY 14853, dmk2@cornell.edu

These are admittedly lofty goals, especially for a first-year faculty member creating a single-semester course. To maximize the potential impact, the first author applied for a Cornell University Innovation in Teaching grant [1]. The program that awards these unique grants is part of a larger Distributed Learning initiative supported by the President and the Provost of Cornell University. These grants are intended for faculty members who have innovative ideas for substantially improving an educational process by leveraging the impact of contemporary information technologies in their teaching. This award gave rise to three more course goals that support the original three:

- to expose the students to technologies (for collaboration, communication, project management, or engineering) that are likely to be *similar* to those they will need to use in their careers,
- to leverage technology to reinforce the course's emphasis on developing excellent communication skills, and
- to utilize available technologies to evaluate the success of the students' learning and of the general course design.

IMPLEMENTATION

The Innovation grant has shifted the nature of the course from its original conception: on one hand, if the new goals are met, the course will be all the more effective in fulfilling its original goals; on the other hand, meeting the additional goals presents new challenges and responsibilities that are addressed largely by the second author, with the collaboration of the course instructor and many talented people from Cornell's Center for Learning and Teaching and its Academic Technology Center. What began as one person's "what if?" has become a collaboration among people with diverse backgrounds, and the students themselves have become co-creators of the course. "Real-World Engineering" has indeed become a microcosm — in more ways than originally anticipated — of a dynamic, industrial engineering and business environment. And the experiment seems to be working.

As it turns out, the course is not currently set up to fulfill the design project requirement for the MEng degree, so the first goal has been temporarily set aside. Nonetheless, several of the students are doing "extra" projects related to the course, but which are outside the scope of the course project itself. Future offerings of ECE 595 may allow students to fulfill their MEng design project requirements within the scope of the course.

Addressing the second goal in a single semester presents an interesting challenge. Learning about hardware design, computer architecture, operating systems, or compilers alone can take several (very intensive) semesters. In order to provide the students with as close to a "real-world" experience as possible, we chose an embedded processor (which is simpler than the general-purpose processors in most desktop machines, for instance) and a small operating system that interacts with that processor to run user programs. These happen to be the eight-bit Hitachi H8 series microcontroller (the H8/3297 running at 16MHz) found in the RCX (or programmable "brick") of the LEGO® Mindstorms Robotics Invention System 2.0 by LEGO Systems, Inc., running the brickOS [2] freeware, an alternative operating system for the RCX.

Addressing the third goal has proved an even bigger challenge. Requiring frequent writing assignments puts an exceptionally heavy burden on the instructor and teaching assistant(s). The solution thus far has been to have the students evaluate each other, giving constructive feedback both in person and anonymously through web-based discussion boards. Each week the role of Project Leader rotates to a different member of the team. Each individual team member sends his or her status report to the Project Leader, who organizes and summarizes the information, and forwards the group report to the "CEO" (the instructor). To build oral presentation skills, the students pair up to do fifteen-minute presentations on short books or book sections (where the content of the book focuses on building effective teams [5]-[7], surviving impossible projects [8], turning companies around in difficult times [9], or building robust engineering systems[10]). The presentations may take any form, but must be collaborative. They are videotaped and later made available to the class via a streaming media server (as well as in standard videotape format). The motivation for this organization is manifold: students who must miss the presentation due to other obligations can watch them later; not all students are required to read all books on the "reading list" (having once been a student herself, the first author assumed that many students would skip reading several books), but can decide based on the presentations whether the book would be of particular interest to them; and the greatest advantage is that the students can watch themselves as presenters, giving them a perspective they would not otherwise have.

The course is thus logically divided into two parts: one technical, one non-technical. The trick is to design both halves to reinforce each other, which should be much easier to do for future offerings of ECE 595, once we have feedback from the current students who are our co-creators of the course.

HARDWARE AND SOFTWARE

The LEGO® Mindstorms Robotics Invention System 2.0 includes a USB infrared (IR) transmitter ("the tower") used to program the RCX bricks. Robots may be constructed from the set's 718 LEGO pieces, which include a light sensor, two motors, and two touch sensors. The sensors included in the kit are not always sufficiently precise for a serious robotics project, but fortunately other manufacturers make more sophisticated sensors and special purpose devices (e.g., compasses) to use with the RCX brick.

The brickOS is an open-source, embedded operating system providing a C and C++ programming environment for the LEGO Mindstorms Robotics Kits. It runs applications developed in C or C++ using the GNU [3] freeware compilers (cross compilers for the Hitachi H8). The brickOS package includes tools that allow the user to download the OS and compiled application programs to the RCX via the IR tower. As a whole, the package provides a useful alternative to the standard LEGO Programming Language, which was designed to be part of a child's toy (albeit a sophisticated one). Originally developed on Linux, the brickOS tools now run on most Unix and Windows platforms. Other C-like ("NQC", or "not quite C") and Java-based programming environments (e.g., LeJOS [4]) also exist for the Mindstorms.

The brickOS distribution provides the sources for the operating system, a handful of demo programs, and the utilities. Instructions on the brickOS web pages help the user to obtain and configure the necessary tools, all of which make it possible to control robotic creations in ways that the original Mindstorm systems do not enable.

STUDENT PROJECTS

The course enrollment was limited to about 20 students (some who had originally signed up found they had scheduling conflicts, and a few others joined late). To give the students the opportunity to exercise the leadership and team communication skills stressed in the non-technical portion of the curriculum, we intentionally threw them into a chaotic environment. We set broad goals, but we gave them little specific instruction beyond the setup of the hardware and software, and help with performing initial sensor experiments. Figure 1 illustrates one of the robots in use in the lab.

We split into three teams: an "Infrastructure" team responsible for designing, building, and testing a radio packet communication system for the RCX with brickOS, and two teams designing and building a pair of robots each such that these "bot" pairs can compete against each other in a Capture the Flag game. The students chose to implement the Capture the Flag competition because most had implemented a simulated version in an earlier engineering course. They have defined the specifications for the game itself, the playing field, and the communication infrastructure; in essence, they design their own labs, and conduct the necessary experiments (which is appropriate for a graduate-level course).

The game consists of two teams, each with two players, and each having a flag. In addition to the bots from both teams, the game board contains static obstacles to be avoided. Each team tries to capture the other's flag and carry it to its "home". The bots start at known positions, and the flags initially reside at known, base positions. Collisions cause a flag-holding bot to relinquish the flag, which returns to its base. A bot holding a flag must announce its position, and general fair play is assumed.

Normally, Mindstorm robots communicate only through their infrared (IR) ports, which requires that two communicating robots be positioned so that their IR ports line up. This would be difficult to do in an implementation of Capture the Flag, plus one of the original goals of the course was to give students an opportunity to build hardware devices. The Infrastructure team is building an interface that uses the LEGO RCX IR port to transmit an analog signal (using the LEGO network protocol, "LNP") to an ATMEL AVR chip [5], which outputs a digital signal to a Radiometrix chip that reliably sends the radio packet payload (including detecting and compensating for collisions). Implementing this new communication channel requires modifications to the brickOS (to prevent reading from and writing to the ATMEL simultaneously). The Infrastructure team has defined an API for Team 1 and Team 2 to use in programming their bots.

FIGURE 1
SAMPLE ROBOT

BUILDING COMMUNITY

Early in the semester, while we were still defining details of the competition rules, determining how we might build the playing field, and estimating total project budgets, we held two "company retreats" at a local restaurant: part of the time was spent socializing and getting to know each other better, and part was spent making presentations with laptops and a portable projector.

Guest speakers from industry come in to provide a diversity of perspective on the career paths available to an engineer. These speakers share their own experience, field excellent questions, and provide a wealth of mentoring advice. Since these presentations happen in the evenings, largely at the convenience of the speaker, not all students are able to attend all events. The presentations are videotaped and made available from the course websites so that students may view or review them, and so that future offerings of the course may derive some benefit from previous semesters' speakers (although the course emphasizes the importance of face-to-face meetings whenever possible). This term's guests include one person whose career has taken him from academia to industrial research to management at a large corporation; one who chose the MBA and startup company route after a BSEE degree followed by several years at a large corporation; one who earned an MEng degree at Cornell before continuing for a Ph.D. and a career as an academic researcher and inventor; and one who, after completing a masters in engineering, spent time designing hardware, building hardware design software (CAD tools), marketing that software, writing speech recognition software, and most recently participating in diverse projects as a Research Scientist in an academic environment.

Finally, some of the course lectures have been organized as discussions about interviewing skills or how to construct a more effective resume.

All three of these activities — the "retreats", the discussions with external mentors, and the class discussions in which all share appropriate experiences to help the others in the group — provide the students in this pilot class with the foundations of a "human resource network" they can leverage and support throughout their careers, whichever direction they may take. What the students have not yet been told is that they will be asked to write themselves a letter at the semester's end and then to keep in touch with the course instructor (at least to the point of updating contact information). One copy of the letter will be sent to the author after two years, and another copy after five. Each copy will accompany a request for any additional feedback (if any) about the course. This activity should help maintain the "network", and should provide a mechanism for evaluating the course's long-term impact.

OTHER TECHNOLOGY CHOICES

In order to select appropriate technologies for this and future offerings of "Real-World Engineering", the Academic Technologies Center and Center for Learning and Teaching have helped to define a concrete set of desired learning outcomes. These are described in more detail in the next section.

The first challenge was to par down the number of technologies to which we wanted to expose the students. We omitted calendering programs, work-flow management software, and specific project-management tools from our roster. These can be incorporated later, once the course content has solidified, and the course goals have been refined by experience. Similarly, we omitted teleconferencing, due to the logistical difficulties and potential expense.

We included CourseInfo [11] for its (anonymous) bulletin board and file exchange features, as well as for the ease with which the streaming videos could be made available to students while being password-protected from the outside world. While not all features of this software product seemed appropriate for technologically savvy engineering students, we decided to evaluate its usefulness within the context of the course.

We selected Dreamweaver [12] to help with web page development and maintenance (and the portfolios described below); this decision was based on ATC's previous experience with the product and the availability of the software on computer labs throughout campus.

As the Innovation grant comes to an end, the instructor and teaching assistant(s) will become responsible for videotaping course activities, with ATC supplying the necessary training and equipment (including recording and dubbing) for ECE and the instructor.

For evaluation, we are relying largely on WebSurveyer [13] and individual student comments. The former helps quantify how well we are achieving the desired learning outcomes, and the small size of the course makes the latter not just possible, but invaluable.

LEARNING OUTCOMES

Having selected a few technologies with which to begin, we then enumerated the remaining learning outcomes that we hoped to achieve, and classified them first as team or individual activities, and then as artifacts or personal reflections. Table I shows this taxonomy. Note that these are based on the initial goals for the course and on the selected technologies; experience with running the course and with new technologies will undoubtedly change how some outcomes are defined, documented, and evaluated.

In addition to the final Capture the Flag tournament, we have asked the students to build a web portfolio per team. These portfolios will contain team artifacts (e.g., the weekly status reports, the project budget, and photos of their robots) that may be annotated with additional information (e.g., details of specific experiments that influenced design decisions). The portfolios expose students to different tools from what they have used before, reinforce the cooperative nature of the project, and provide a record of what the team accomplished and how.

The portfolios may eventually be used to help advertise the class or the individual students (e.g., it could

TABLE I
DESIRED LEARNING OUTCOMES

Learning Outcome	Assessment Method or Object of Record	Artifact or Reflection?	Team or Individual Product?
Work productively as members of an effective team	weekly individual progress reports weekly team progress reports	artifact artifact	individual team
Act as Project Leader and demonstrate/improve project management skills	weekly team progress reports individual final paper	artifact reflection	team individual
Learn to manage conflict in teamwork situations	lab notebook on team dynamics weekly team progress reports individual final paper	reflection/artifact artifact reflection	individual team individual
Learn to impose structure on an unstructured engineering design problem, translating end goals into a procedure for accomplishing them	lab notebook documenting student-designed lab experiments team project summary of project process	artifact artifact	individual team
Design experiments that effectively answer engineering questions	lab notebook documenting student-designed lab experiments team project summary of project process	artifact artifact	individual team
Design elegant solutions to engineering problems (based on experimental results)	lab notebook documenting student-designed lab experiments team project summary of project process	artifact artifact	individual team
Learn to respect different opinions	lab notebook on team dynamics individual final paper	reflection reflection	individual individual
Demonstrate improvement in written communication skills for business	weekly individual progress reports weekly team progress reports final team portfolio	artifact artifact artifact	individual team team
Demonstrate improvement in oral presentation skills	videotapes of student presentations feedback from class members individual final paper	artifact artifact reflection	individual individual individual
Give constructive feedback on others' work	oral feedback to others (on video) written feedback to others (on class discussion board)	artifact artifact	individual individual
Encounter and consider diversity issues (related to culture, gender, personality types, or work habits) within teamwork situation	lab notebook on team dynamics team self-evaluation surveys (and follow-up with instructor)	reflection/artifact reflection/artifact	individual individual/team
Demonstrate active contribution to final output of team	weekly individual progress reports	artifact	individual
Document a design process	final team portfolio	artifact/reflection	team

be something they list on their resumes). Similarly (but separately from the portfolios), video clips of student presentations and the robot competition will be assembled over the summer to create a short video to summarize and advertise the course and the Cornell MEng program. Once the course is established at Cornell, we hope to export its infrastructure and ideas to other institutions (perhaps even adapting it for undergraduate or K-12 use) if there is sufficient interest.

The students have been asked to keep a two-sided "lab notebook" throughout the semester, where one side records experiments, design options, budget information, and the like. The other side is intended to record experiences with team interaction, different leadership styles, problems encountered and solutions attempted, and personal reflection on the structure or content of the various parts of the course. The "back" part of the notebook will provide the basis of a short paper summarizing what each student found most valuable or most difficult with the course.

Templates for both the web portfolios and the final papers will be provided to the students, but (as always) they may diverge from these to create documents and products in a manner that suits them.

MIDTERM EVALUATION

Table II illustrates the kinds of questions and feedback options that WebSurveyor allows. About half the class has responded to the survey. Of those, all say that the student presentations have been at least of moderate help in their learning; two thirds find the student presentations to be "much help" or "very much help". Responses to the

TABLE II
SAMPLE MIDSEMESTER EVALUATION QUESTIONS

How much are each of the following aspects of the class helping your learning?	very much help	much help	moderate help	a little help	no help	no opinion/not applicable
Student presentations	•	•	•	•	•	•
Discussions in class	•	•	•	•	•	•
Readings outside of class	•	•	•	•	•	•
Hands-on lab activities/experiments	•	•	•	•	•	•
Lab structure (i.e., having students define & run labs)	•	•	•	•	•	•
Teamwork in labs	•	•	•	•	•	•
Lab reports/lab notebooks	•	•	•	•	•	•
Interaction/self-evaluation notebooks/journals	•	•	•	•	•	•
Guest speaker presentations	•	•	•	•	•	•

remaining questions are at least as positive, with the exception of the lab notebook and self-evaluation questions. Since the notebooks and evaluation portions of the course will be used primarily in the final projects (the web portfolio and final papers), it makes sense that some students have not found the tools particularly useful yet. These obviously need to be more fully integrated into the entire semester in future course offerings, since documentation and self-evaluation are import skills to build in any engineering-related profession. Students seem particularly happy with the lab activities, the teamwork aspects of the course, and the guest speaker presentations (scores for these questions gave 89-100% “much help” or “very much help” ratings). Similarly, 89% or more of the responding students give the top two most positive responses to questions about quality of interaction with the instructor, working with peers outside of class, the way the course is taught overall, and their understanding of LEGO robotics and team dynamics. When asked what they are getting from the course (none responded that he or she was not getting what was needed), responses included:

- “The most significant thing I am getting out of this class is the insights into a real work environment in industry and how everything seems to play out there.”
- “The Ability to Express what I do. And to lead a group of people to do a task, however temporary it may be.”

On the negative side, at least one student feels uncomfortable with the lack of structure in the class, and another (legitimately) requests more feedback from the instructor and teaching assistant. Since this course is a work in progress, we will spend the summer designing better ways to achieve our course goals, student learning outcomes, and student needs.

ACKNOWLEDGMENT

This work is supported in part by a Faculty Innovation in Teaching grant from the Cornell Academic Technology Center. David J. Kalousdian from ATC has provided most of the video technology support for this project. Prof. David G. Way of the Cornell University Center for Learning and Teaching has provided invaluable guidance and resources for course development. The Cornell Engineering Library’s ACCEL Computer Lab has provided access to some of the technologies that we’re using.

REFERENCES

[1] McClure, P.A., Cornell University Office of Information Technologies, “Faculty Innovation in Teaching Grant Program”, http://www.cit.cornell.edu/atc/innovation/, 2002-2003.
[2] Noga, M.L., “Welcome to the BrickOS™ Home Page”, http://brickos.sourceforge.net/, 2003.
[3] “GNU’s not Unix — the GNU Project and the Free Software Foundation”, http://www.gnu.org/, 2003.
[4] “LeJOS: Java for the RCX”, http://lejos.sourceforge.net, 2003.
[5] Lundin, S.C., Paul, H., Christensen, J., “Fish! A Remarkable Way to Boost Morale and Improve Results”, Hyperion Publishing, March 2000.
[6] Lundin, S.C., Paul, H., Christensen, J., Strand, P., “Fish! Tales: Real-Life Stories to Help You Transform Your Workplace and Your Life”, Hyperion Publishing, April 2002.
[7] DeMarco, T., Lister, T., “Peopleware: Productive Projects and Teams”, 2nd. Ed., Dorset House Publishing, February 1999.
[8] Yourdon, E., “Death March: The Complete Software Developer's Guide to Surviving 'Mission Impossible' Projects”, Prentice Hall, June 1999.
[9] Gerstner, L.V., Jr., “Who Says Elephants Can't Dance? Inside IBM's Historic Turnaround”, Harper Business Publishing, November 2002.
[10] Brooks, F.P., “The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition”, 2nd. Ed., Addison-Wesley, August 1995.
[11] Blackboard, Inc., http://www.blackboard.com, 2003.
[12] Macromedia, Inc., http://www.macromedia.com/software/, 2003.
[13] WebSurveyer Corp., http://www.websurveyor.com, 2003.