Collaboration between Engineering Departments at Clarkson University for a Freshman-Level Engineering Programming Course Including an Experimental Lab Experience

John E. Bean and John P. Dempsey

Abstract – At Clarkson University, the freshman engineering class, ES100, Introduction to Engineering Use of Computers is a second-semester freshman year course. ES100 was previously handled independently by each of the four engineering departments with little or no coordination between sections. As part of a grant Clarkson University received from the National Science Foundation entitled "Hands-On Engineering" (DUE-0311075), a major effort was undertaken to revamp this important course. The major goal of this project was to increase the number and quality of hands-on experiences in engineering classes and to incorporate academic and industry-standard software packages MATLAB and LabVIEW. These changes required an interdepartmental effort to transform this course from its segmented state to a unified and cohesive freshman experience. The resulting unified course focused on MATLAB, emphasizing programming but also including plotting, simultaneous equation solving and symbolic math. Four laboratory experiments requiring teamwork and report writing were included. Course management was accomplished using the course management system Blackboard. Successes and challenges of two offerings of this course are discussed along with plans for future offerings.

Index Terms – MATLAB programming, Hands-on, CIEC, John E. Bean

Background

In 2003, Clarkson University received a grant from the National Science Foundation entitled "Hands-On Engineering" (DUE-0311075). The primary goal of this grant was to reform the laboratory experience for all engineering undergraduates at Clarkson University, and to vertically integrate core software applications in a number of different courses. Because it was desirable that the advances in research percolate down into the undergraduate educational process, the software used had to be equally viable in both an educational and a research setting. In this regard LabVIEW and MATLAB were ideal choices. MATLAB is especially well suited to be the introductory engineering programming platform [1].

The first class for attempted reform was the freshman engineering class ES100, Introduction to Engineering Use of Computers. ES100 is a second-semester freshman year course taken by approximately 350 students, mostly engineering majors. Prior to the Spring of the 2004-2005 academic year, this course was handled independently by each of four engineering departments, each with their own instructor, teaching solely to their own majors, using their own, independent syllabus and typically, taught in a lecture hall format. There was no coordination between sections, no required standardization of computer languages or tools and no shared vision for the outcomes of the course. None of the sections included any laboratory experience outside of computer lab. The result was a wide disparity in class sizes (ranging from 30 to 175), material covered, tools and concepts learned, quality and academic rigor. There was no common computer literacy that could be assumed across disciplines or even from year to year within the same major. The reform of this class involved all four engineering departments and was actively supported by the administration of the Coulter School of Engineering. Two teams of faculty and graduate students were formed: one to work on the overall curriculum issues and one to develop multi-disciplinary laboratory experiments.

Lecture

The team organized to develop the curriculum for the class consisted of the principal investigators for the project, a graduate student (first author) and two professors who were teaching the course at the time. It was an excellent group, combining the real-world wisdom of those teaching the course with those who had envisioned the changes to a more “hands-on curriculum” and received the grant funding from the National Science Foundation.
ES100 is a two-credit course. There was significant discussion at the outset about adding another credit in order to make more room for additional content, however, engineering programs tend to be credit-limited and already filled with content. Adding a credit in one course would mean taking a credit away from another course. It quickly became apparent that that would be a lengthy political fight that could not be undertaken if we actually wanted to get anything accomplished with the course and could possibly undermine support for our larger goals. We had to make hard decisions about what we could accomplish within the existing two-credit framework of the course.

At approximately the same time this was taking place, a memo came out of the Mechanical and Aeronautical Engineering Curriculum Committee that strongly supported the reform of ES100, in particular, the programming content of the course. It stated “It is generally accepted by the faculty in the school of engineering that the programming skills of our students could use substantial improvement”. A consensus was thoughtfully but quickly achieved that ES100 would focus on the programming skills of our students. We would use MATLAB as the programming platform and LabVIEW to control the accompanying experiments (although no LabVIEW programming is included due to time constraints). The experimental data generated and collected would then provide “real-world” data for the students to process and display using MATLAB.

A topical outline was developed and an appropriate text was chosen for use in all sections. Since the emphasis was to be on “hands-on” learning, it was agreed that we should have class in a computer equipped classroom so that students could try things as they were being discussed rather than listening to a lecture about programming then waiting for a period before attempting the homework. Unfortunately, the School of Engineering did not have the capacity to handle this number of students and sections in computerized classroom but the school of business did. Their generosity in scheduling allowed this to become a reality. We ended up making twelve sections of 25-35 students each as opposed to the four sections ranging from 35-175 students each. The students were randomly assigned to sections so there would be a heterogeneous mix of majors in each section.

Course Management

In order to handle the management of this course, the Blackboard Course Management System (Bb CMS) was employed. This CMS was already in-place at the university and used to some degree in a few classes. We utilized Bb heavily. We developed our own Powerpoint slides, exercise files, help resources, homework assignments and lab instructions; all made available via Bb. Homework files are all submitted digitally using the “digital dropbox” feature. Every class begins with a short online quiz run from Bb which is graded automatically, provides feedback and stores the grade. Three summary quizzes (cumulative quizzes from previous classes) are spaced through the semester. Each class ends with a “muddiest point”, an assessment technique asking students what they considered to be the “muddiest” point in the lecture, homework or reading [1]. The muddiest point is implemented in Bb as a quiz with an essay question. It has proven to be very useful to the instructors as immediate feedback and forces the students to revisit the lecture in their heads. Some of the students tried to short-circuit the process by answering “no muddiest point”. We now require that if there is no muddy point, they must list the most interesting point. The muddiest point is not graded per se but completion is required to receive credit for class attendance.

One of the difficulties encountered in using Bb was the logistics of multiple sections of the same course. At least in the “basic” version the university is presently using, there is not an ability to handle multiple sections easily: either all students are in one big section or separate sections need to be set up and updated individually. Since we had twelve sections taught by four instructors for this one course, no single solution would be without drawbacks. For the first offering we used a single Bb section. For the individual maintaining the CMS this was the easiest approach to deal with; all students accessed the same materials, announcements went out to all students, etc. However, it was very difficult to pull out information for a student by actual section or instructor. Often the Bb data would be downloaded to an Excel file and brought into a Microsoft Access relational database to facilitate rearrangement by instructor and/or section. During the second offering of the course, four Bb sections were used; one for each instructor. The instructors generally liked this much better but maintenance was problematic and there was a lack of
consistency between instructors; some inundated their students with announcements, rearranged categories and items while other instructors basically took a hands-off approach. It has become clear that either way it is handled, there needs to be procedures developed to ensure consistency for the sake of the students and for the person(s) maintaining the system.

**Laboratory**

Another of the desired changes to the course was to incorporate laboratory experiments that could be performed by the students to further pique their interest in engineering and to generate real-world data sets for processing, analysis and reporting. From a survey of national curriculum reform efforts, it became clear that the ITL program initiated in 1992 at UCBoulder could be adapted and implemented to satisfy the goals and objectives of this proposal. In what was then a very ambitious vision for undergraduate engineering education reform, UCBoulder sought ‘...to pioneer a multidisciplinary learning environment that integrates engineering theory with practice and promotes creative, team oriented problem-solving skills’ [3]. The motivation for change in engineering education at UCBoulder was driven by ‘the understanding that traditional, strictly theoretical, engineering education does not engage or adequately educate engineering students.’ [4] The ITL Program at UCBoulder focuses on preparing students to work in a team environment and to understand and experience the problem-solving process. [3]

A team of faculty and graduate students were tasked with designing a multidisciplinary experiment that would include aspects of each area of engineering involved. It was required that the system utilize a computer-based data acquisition system and be controlled using LabVIEW software in order to provide the students with exposure to these tools and technology to provide a foundation for higher-level classes where these are used.

It was desired that the experiments involve a system that could be tested in component parts and as a whole to emphasize breaking down a complex problem into its component parts for better overall understanding. In addition to these requirements, and the need for the experiments to be interesting to freshman students, the experiments needed to be economical, rugged, reliable and straight-forward as well.

**Overview of the Experiment**

The experimental system developed was designed around an alternative energy system, i.e., a fuel cell apparatus. This setup included a solar panel, an electrolyzer, two gas holding tanks, two fuel cells and a variety of load devices, e.g., thermoelectric cells, small motors, etc. The laboratory experiments were broken into four parts: solar panels, electrolyzers, fuel cells, and then a complete system including loads. Each of the first three laboratory experiments was designed to measure the operating characteristics of one of the fundamental components in the system. Once the behavior of each component was understood, the overall system was examined for a given load.

For example, during the electrolyzer experiment, students used LabVIEW to record voltage and current waveforms applied to the electrolyzer. Using these data and information gathered from visual inspection of the gas holding tanks and time, students learned how much hydrogen gas the electrolyzer could produce based on its power input and its efficiency. They also investigated the chemical reaction of water and how it applied to the electrolyzer. A thermocouple was embedded in the electrolyzer to monitor temperature as the reaction took place.

There are eight lab stations consisting of an IBM Intellistation M Pro computer with a National Instruments data acquisition card. Each station is equipped with an Agilent 344014A 6 ½ digit multimeter and an Agilent E3631A Triple Output DC Power Supply as well as an auxiliary power supply and an Elenco Resistor Substitute Box to facilitate the aforementioned energy system experiments.

The class of 350 students was broken up into twelve (12) sections (different from the lecture sections). The laboratory experiments were performed by teams of, typically, four students who were required to run the experiment, download the data files to be processed and analyzed in MATLAB and to submit a written report. Students were required to evaluate their own, as well as their fellow student's, performance.
Conclusion

The response of the students has been mostly positive; many have stated that the use of "real" data makes the programming concepts taught in lecture much more meaningful to them. The students like the consistency between sections of the course but some feel lectures are driven too much by the Powerpoint slides. For many of these students, this experience involves the most intensive teamwork they have had to deal with in their educational experience. They learn both the positive aspects of teamwork as well as the liabilities. A student survey has been developed to gauge student opinions on all aspects of the course, including the lab experiences. Adequate data for preliminary analysis of student attitudes is now available. One of the key outcome indicators will be feedback from instructors of subsequent courses where the skills and knowledge learned here should have a major impact on future student performance.

Acknowledgements

The support of NSF's Division of Undergraduate Education via Grant DUE-0311075, and NSF's Antarctic Ocean and Climate System Program, via Grant OPP-0338226, is gratefully acknowledged.

References