WIRELESS, WEB-CONTROLLED, BALL-COLLECTING ROBOT: AN ENGINEERING AND COMPUTER SCIENCE CLUSTER COURSE

Todd Swift\(^1\) and Danial Neebel\(^2\)

Abstract — A senior engineering design and a computer science capstone course are clustered for an interdisciplinary experience. In the fall of 2002, students designed and implemented a wireless, web-controlled, ball-collecting robot. A single computer, configured as a web server, controls access and wirelessly sends commands to the robot. Students developed a full set of specifications for the robot and web server system and maintained a project management plan and schedule throughout the semester. Unique aspects of the course include an electronic project log that is shared using a course information system and a peer evaluation system that constitutes a significant portion of the final grade. The course allows students to explore their disciplines further, gain experience working in interdisciplinary teams, and learn the importance of technical communication via a final design report and presentation. Nearly all of the students reported that they learned more working in an interdisciplinary setting than they would have working only in their discipline.

Index Terms — Clustering, interdisciplinary, mobile robotics, senior design projects.

INTRODUCTION

Much has been written about the advantages of encouraging students to work on problems from an interdisciplinary perspective \(^8\) and many real world problems are too complex to approach from a single discipline. The clustering of courses from two disciplines is one method to facilitate an interdisciplinary experience for students. The Accreditation Board for Engineering and Technology even lists “the ability to function on multi-disciplinary teams” as one of the 11 main required program outcomes \(^1\).

The authors have clustered a senior engineering design and a computer science capstone course for an interdisciplinary experience. In the fall of 2002, students designed and implemented a wireless, web-controlled, ball-collecting robot that is able to avoid colliding with obstacles and is controlled by external users via a web server.

In this paper, we will provide background and a history of the clustered course at Loras College, describe several unique aspects of the course, summarize the results of this year’s project, and share assessment efforts that we have used to improve the clustered course process.

COURSE ORGANIZATION

The Loras College Department of Computer Science has required a capstone team design project of its seniors since 1986 \(^7\). In 1997, the Department of Physics and Engineering at Loras College developed a new program entitled Electromechanical Engineering. In an effort to make both programs interdisciplinary, the required computer science senior capstone course was clustered with the senior engineering design course. See \(^6\) for a description of the earlier course and projects.

Since 1998, the projects have involved mobile autonomous robots including line-tracking race cars \(^6\), fire-fighting robots \(^9\), and this year’s project, a wireless web-controlled ball-collecting robot.

Instructor’s Role

Instructor’s Role

Previous implementations of the course were fairly prescribed in that after selecting the project, appropriate steps were selected by the instructors and built into mini-projects complete with reports and presentations \(^6\). This method has the advantage of keeping the course very organized and maintaining a prescribed schedule, but, in the authors’ personal professional experience, real-world design teams do not have such prescribed mini-projects and must determine the best course of actions and the critical timeline for a certain goal as a team.

In an effort to duplicate a less-prescribed experience, we restructured the course to minimize instructor micro-management of the project. Our role was primarily as ‘managers’, and we shifted more of the overall project schedule and maintenance to the student teams.

Course Format

We met with the senior students near the end of the spring semester their junior year to share with them the goals of the course, let the two teams meet each other, and gather their input on their project. We had a list of requirements for the project, but the students had significant input on their project and were asked to brainstorm and research ideas before the meeting. Our major requirements are straightforward:

- The project must involve the Handy Board, a Motorola HC11-based microcontroller developed by Fred Martin and MIT’s Media Lab \(^5\). This requirement is due to

---

\(^1\) Todd Swift, Loras College, Department of Physics and Engineering, 1450 Alta Vista, Dubuque, IA 52001 toswift@loras.edu

\(^2\) Danial Neebel, Loras College, Department of Mathematics and Computer Science, 1450 Alta Vista, Dubuque, IA 52001 dneebe1@loras.edu
our current stock of Handy Boards and our past success with them.

- The project must have a significant mechanical design component.
- The project must have a significant software design component.
- Several feasibility requirements (1 semester to complete, fit within our budget, within the technological capabilities of the group, etc.)

Although in general, we do not get 100% agreement on the selected project, students are happy that they have input and do have ownership of the project.

At the first meeting of their senior course, we presented what we called a User Needs Document in which we described, in fairly vague terms, what we wanted the robot to do. The User Needs Document was purposefully vague and incomplete since in the authors’ professional experience, product users and management and marketing staff often do not specify product features to the degree of precision required for design decisions.

We then gave the student team about four weeks to develop a complete Product Specification in engineering and software development terms. We wanted the students to tell us (“management”) exactly what they would build and exactly what it could do including the software interface, dimensions, storage capacity, etc. We found that producing an acceptable Product Specification required a lot of feedback and guidance from us. Students did not provide the level of detail that we desired without our assistance.

Along with the final Product Specification, students were required to present a Gantt chart highlighting the critical path in the project and proposing schedules for each subcomponent of the project. This is difficult for students. It is also difficult for practicing engineering and software design teams!

We then based the remainder of the course schedule on the Gantt chart. Students gave us weekly progress reports and preliminary designs that were turned in for feedback and grading and the Gantt chart was continually modified and updated. Some action items took longer than expected and items that were not even listed as major-subcomponents were discovered.

Students received graded feedback on several reports through the semester, but the majority of weekly feedback came from an electronic project log that is described below. At the end of the semester, the students gave a campus-wide presentation and prepared a final project report.

One additional aspect of the course that we will discuss below is that engineering students receive one additional academic credit for the course (4 credits) compared to computer science students (3 credits). The extra contact hour is used for professional development issues: job hunting, resume writing, practice interviews, etc.

Electronic Project Log

Our campus recently adopted full-scale use of Blackboard [3], an electronic course-management system. In addition, all full-time students at Loras College have possession of a leased laptop computer. Blackboard was used extensively for document sharing and for the first time was used to log all student work in electronic logbooks.

We created a blank log form in a word processor that lists date, starting time, ending time, total time, and team members present (including the name of the note taker). The remainder of the form has space to detail any work that is completed during a meeting. Each time any members of the team met, they were to fill out this form in real time. We made a digital camera and a scanner available to students so that pictures and scanned sketches or references could be included as needed.

We required an up to date log be maintained at all times and available for all at the course Blackboard site. Accessing the documents to provide feedback became very easy. We used the ‘track changes’ feature in our word processor to comment the logbooks, which we could then repost immediately and use to lead discussion during class time. We would contact the team leader if the logbook was not kept current. This allowed us to have one contact person so we did not have to rely on contacting multiple students. Leadership was rotated throughout the semester to give all students experience in team leadership.

Evaluation System & Peer Evaluation

Evaluation was divided among four categories: reports and presentations describing sub-projects (25%), the logbook (10%), the final presentation and document (55%), and class attendance and participation (10%). The scope of main sub-projects was largely determined with student input.

A unique aspect of the evaluation of the course is a peer evaluation system that we call the Individual Performance Measure. Each student is asked to distribute 100 percentage points among every student in the course (both disciplines together). Fifty percent of their final grade is then weighted by the sum of this factor for each student, although we do reserve the right to adjust any peer grading that is obviously self-serving, malicious, or largely inconsistent. In addition, we leave space on the form for students to assign a letter grade to other students and to explain why they rated them as they did. A copy of the form is included in Appendix A.


The initial problem statement for the 2002-2003 project was: “Design and build a robot that cleans up ping-pong balls in a small room. While the robot must be able to move around the room, it is not completely autonomous in that it must receive...
commands from a human user via the WWW. The commands will come from a remote computer that is running a web server and allows users to control the motion of the robot via the Internet. The user issuing commands will be able to receive feedback on those commands via a camera connected to the web server showing pictures on the web site.”

In addition to the above problem statement, students were given a list of about a dozen constraints regarding such things as size, minimum number of balls to be carried on the robot, safety issues, and other constraints chosen to match with our specific educational goals and budget constraints.

As mentioned in the first section of this paper, the student teams came up with the main sub-projects required to complete the robot and software. Main engineering sub-projects included testing and selection of a wide variety of sensors, mechanical design and modeling, and construction, testing, and calibration of the robot after mechanical designs had been approved by the entire team and the instructors.

Main computer science projects were the software design and coding for the server, the client, and the robot. Students chose to use Apache Sever Version 1.3.17 [2] for the web server and selected Java Runtime Environment Version 1.4.1 [4] for the client software. The Handy Board was programmed in Interactive C, which is an interpreted implementation of C that is freely available with the Handy Board [5].

**Project Outcome**

The students successfully completed the design and implementation of the robot. Figure 1 shows a final solid model of the robot while Figure 2 is a photograph of the completed robot. Figure 3 shows the user interface screen on a client computer. The web cam view is in the upper left, the robot controls are in the upper right, and a user list and chat area are on the bottom of the screen. We discuss additional learning experiences and outcomes in the next section.

**EDUCATIONAL OUTCOMES AND RECOMMENDATIONS**

We now take a look at peer evaluation results, student feedback from the course, and our future plans for the course.

**Peer Evaluation Results**

We have used our peer evaluation system successfully for several years and feel that it provides an excellent feedback mechanism for us and for the students. We get to compare the results to our perceptions of how students handled the project and students get to evaluate their peers – which many students are anxious to do by the end of the semester. We have not had a problem with malicious grading and have not had to throw-out or adjust any individuals’ point distribution.

There were 11 students in the Fall 2002 course. The maximum score that any student was awarded by his/her peers out of 100 was 120% and the minimum was 83%. The standard deviation was about 11. Since 50% of the final grade was weighted by this factor, the maximum effect of the factor was about one letter grade above or below the grade the student would have received without this system in place. Four of the 11 students scored over 100 – these were the students who had emerged as leaders of the computer science team (2 students) and the engineering team (2 students). Only one student awarded points equally to everyone and we did not feel that there was any malicious or largely self-serving grading.

An interesting experience this year that had not occurred previously in the authors’ experience is that, due to the specialized nature of the project, computer science students reported that they had a hard time evaluating engineering students and vice versa. During this project, the teams were more isolated from each other than they had been in the past.
An option based on the selected project and team size would be to have students only rate the other students in their discipline. We feel that this would undermine the interdisciplinary nature of the project and the realization that product development depends on the entire team and thus requires some knowledge of what others on the team are doing. In the future, we will work to select projects that require more communication between disciplines.

Electronic Project Logs

We feel that the electronic project logs maintained in Blackboard were successful. Previously, logbooks had been kept in standard experimental notebooks. This causes a problem when two instructors need to read and comment the logbooks while the students are still working on the project. Logbooks cannot be kept in real time when instructors have the notebooks! This caused the quality of the logbooks to suffer in the past.

Keeping the logbooks in Blackboard allowed us to easily comment them and get them back to students immediately by reposting the commented logs to the course Blackboard site. Students could meanwhile continue to add on to the master log file. We did find that the electronic files often became very large and thus we do suggest breaking them into separate files periodically to manage the size of the posted files. The separate files are easily appended to a master log file with all instructor comments embedded.

Students initially had a difficult time providing the detail that we required to make the logbooks useful, but by semester end reported that they liked the electronic log book and were very proficient scanning rough sketches, photographing chalkboards, and keeping a detailed notebook. A big challenge in this area is to convince students of the necessity of detailed documentation. Students tend to learn this the hard way, but by the end of the semester they did admit that without the detailed logbooks, they would have had to repeat several experiments and/or calculations. We fully recommend the use of electronic project logbooks as an alternative to traditional written experimental notebooks in a situation similar to ours.

Course Survey Results

In addition to the peer evaluation, students were given a survey at the end of the course asking them several questions including the maximum and minimum number of hours they spent on the course in any one-week period, how they felt about the workload and distribution of work in the course, how well the team and sub teams worked together, the value of the interdisciplinary nature of the course, suggestions for improvements to the course, and a specific question regarding the amount of direction that we gave to the projects in the course. All but one student returned a completed survey at the end of the course. A copy of the survey with descriptive statistical results is included in Appendix B.

Students reported spending an average of 12 hours per week on the course. The large standard deviation of 6 hours suggests that the load was not equally carried by all of the students as we had observed. The maximum number of reported hours spent in any one week by an individual student was 40 hours. The students did feel (with a small standard deviation) that the workload for the course was ‘about right’ on a 5 point scale ranging from ‘much too light’ to ‘much too heavy’. Interestingly, students either overestimated time that they spent on the course each week or did not log all of their time, as the average number of hours per student per week obtained by adding up the total amount of recorded time in the log book was about 7 hours.

Another central question to us was how the students perceive the value of the interdisciplinary nature of the course. We asked them how the interdisciplinary nature of the course affected their learning during the course. On a five point scale ranging from ‘much more’ (1) to ‘much less’ (5), with 3 equal to learning ‘about the same’, students reported an average of 1.9 with a standard deviation of 0.6 suggesting that they felt they learned more in the course due to its interdisciplinary nature.

Also of major interest to us was how the students felt about our role in the course. We acted as managers, but were intentionally more “hands-off” than in the past. Although students did report frustration with this at times, on the end of the course survey, 9 of the 10 students that responded felt that they would have learned less if we had more actively managed the course. The 10th student was neutral.

Additional questions on our survey included several questions about team dynamics and workload distribution between the teams. See Appendix B for these additional questions and survey results.
Additional Results

The engineering students earn 4 credits for their course and the computer science students earn 3 credits. The “4th credit” in engineering is used for professional development. The students write a resume and cover letter and even go through a practice interview with a practicing licensed engineer from a local consulting firm.

One side effect of a course clustered across disciplines is that the instructors get a chance to learn what is being done in other areas. The computer science program is now revising their curriculum and considering ways to include professional development. As a way to encourage internships, the computer science program plans to require resumes and cover letters for courses that occur earlier in the curriculum.

Finally, computer science students are required to take a one-semester capstone course while engineering students take two semesters of the senior design course. This has resulted in some friction between students during the end of the fall semester. Some computer science students felt that engineering students lacked motivation during the end of the semester since they could finish any remaining work during the spring semester. It would be ideal if both programs required the same number of semesters for their senior projects.

ACKNOWLEDGMENTS

The authors would like to thank Steve Mosiman (Department of Mathematics and Computer Science, Loras College) and Christoph Heimcke ( Fluent Inc., Lebanon, NH) for their previous efforts in clustering these courses and for recommendations and feedback as we constructed our course. We would also like to thank the other members of our departments for their flexibility in working around the difficulty of concurrently scheduling senior courses from two departments.

REFERENCES

[9] Trinity College Fire-Fighting Robot Contest, accessible online at http://www.trincoll.edu/events/robot/

APPENDIX A: PEER EVALUATION FORM

All members of the project team must complete and turn in this team member evaluation. You may use the table on the next page. Your information will be used to determine the project grade for all projects and all individuals. The percent effort for each person will be applied to 50% of the course grade.

1. In the second column of the table assign a percent of the total effort to each person on the project team. The sum of all numbers in the second column must be 100. There are 11 people on the team. If everyone put in the same amount of effort, everyone would have a percent effort of 9.091%.
2. In the third column give each person a letter grade, where A shows superb work and F shows no work.
3. Answer the questions on the bottom of the next page. You may add an extra sheet if necessary.
4. Turn in your evaluation sheet by Dec. 6th.

Example Sheet:

<table>
<thead>
<tr>
<th>Project Member Name</th>
<th>Percent of Effort on Project</th>
<th>Letter Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nancy</td>
<td>38</td>
<td>A</td>
</tr>
<tr>
<td>Fred *</td>
<td>37</td>
<td>A</td>
</tr>
<tr>
<td>Eminem</td>
<td>25</td>
<td>C</td>
</tr>
<tr>
<td>Total Work</td>
<td>100%</td>
<td>----------</td>
</tr>
</tbody>
</table>

Fred turned in the above example evaluation sheet. If Eminem received a rating of 22% from Nancy, 25% from Fred, and 33% from himself, his individual performance measure (IPM) would be (22+25+33) = 70%. To see how the final grade is calculated using the IPM, see the course syllabus.

<table>
<thead>
<tr>
<th>Project Member Name</th>
<th>Percent of Effort</th>
<th>Letter Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student #1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Student #2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Student #3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Student #4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Student #N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Work</td>
<td>100%</td>
<td>----------</td>
</tr>
</tbody>
</table>

Strengths: For each individual that you ranked highly, either with a large percent or a high grade, please tell why you gave that individual that particular rating.
**Weaknesses**: For each individual that you ranked poorly, either with a small percent or a low grade, please tell why you gave that individual that particular rating.

**APPENDIX B: COURSE SURVEY**

[NOTE: The number of responses for each survey choice is given in brackets after the item.]

1. What is your major?
   i) Computer Science [6]
   ii) Engineering [4]

2. How many hours a week on average did you spend on the course outside of the classroom? [Range = 2 to 22.5, Average = 12, Standard Deviation = 6.0]

3. What is the maximum number of hours that you spent on the course outside of the classroom during any one-week period? [Range = 10 to 40, Average = 23, Standard Deviation = 10]

4. What is the minimum number of hours that you spent on the course outside of the classroom during any one-week period? [Range = 0 to 15, Average = 5.1, Standard Deviation = 3.9]

5. How do you feel about the workload in the course?
   i) Much too heavy [0]
   ii) Too heavy [1]
   iii) About right [9]
   iv) Too light [0]
   v) Much too light [0]

6. How do you feel about the distribution of work in the course?
   i) I did much more than my share of the work [2]
   ii) I did more than my share of the work [2]
   iii) The work was distributed about evenly [4]
   iv) I did less than my share of the work [2]
   v) I did much less than my share of the work [0]

7. How well do you feel that your (engineering or computer science) subgroup worked together?
   i) Very well [2]
   ii) Well [6]
   iii) Ok [1]
   iv) Poorly [1]
   v) Very poorly [0]

8. How well do you feel that the whole group worked together?
   i) Very well [1]
   ii) Well [3]
   iii) Ok [5]
   iv) Poorly [1]
   v) Very poorly [0]

9. How do you feel about the workload distribution between the two disciplines (engineering and computer science) over the course of the whole semester?
   i) The engineers did much more work than the computer scientists [1]
   ii) The engineers did more work than the computer scientists [1]
   iii) The work was about evenly distributed [7]
   iv) The computer scientists did more work than the engineers [2]
   v) The computer scientists did much more work than the engineers [0]

10. How do you think working with two disciplines affected your learning experience?
    i) I learned much more working with both disciplines [2]
    ii) I learned more working with both disciplines [7]
    iii) The amount of learning was about the same [1]
    iv) I learned less than I would have if I had been working only in my discipline [0]
    v) I learned much less than I would have if I had been working only in my discipline [0]

11. How well do you feel that your previous coursework prepared you for your senior project?
    i) Very well [3]
    ii) Well [3]
    iii) Ok [3]
    iv) Poorly [1]
    v) Very poorly [0]

12. This year, you largely controlled the direction of your project. Would you have learned more or less if there had been more active management of the sub-projects in the course? Please be specific. [Learned less = 9, Learned same = 1, Learned more = 0]

13. Overall, how do you feel about your experience in the course? [One student did not report]
    i) It was very valuable [5]
    ii) It was valuable [2]
    iii) It was ok [1]
    iv) It was not very valuable [1]
    v) It was mostly a waste of time [0]

14. What suggestions do you have on how to improve the course?

15. What are the chief strengths of the course?