Robotics in Education: A Tool for Recruiting, Engaging, Retaining and Educating Students

MICHELE VAN DYNE, JACOB FJERMESTAD
Computer Science Department
Montana Tech of the University of Montana
1300 W. Park St., Butte, MT 59701
USA
mvandyne@mtech.edu jtfjermestad@mtech.edu http://cs.mtech.edu

Abstract: - Robotics is being used in the Montana Tech Computer Science Department in a variety of ways: to increase awareness of the field, to assist us in recruiting students, to engage our existing students, and to provide an advanced research and development experience for our upper level students. The use of robotics in our Computational Thinking course has shown statistically significant effect in the tested increase in logical and mathematical thinking as measured with a pre-test and post-test, which has been administered over the last three years that the course has been offered. Other robotics projects that are underway will take time before results are expected, including the implementation of our NAOCam project, since it is still in its design phase. Thus far, though, our results are promising, and we are optimistic that as additional robotics efforts are completed, those results will also be positive.

Key-Words: - Robotics, education, network control, NAO, Lego Mindstorm NXT, computational thinking

1 Introduction

Computer science as an educational discipline has seen a decline in enrollment in the last decade, although the trend appears to be reversing. As the job market continues to be promising for the graduates of the discipline, we see more and more students interested in majoring in computer science. Unfortunately, the Computer Science Department at Montana Tech has found that many who show initial interest become disillusioned with programming early in their education, and there are many more who don't even consider it as a major. Thus, we have problems attracting students since they are unaware of the area, and problems retaining those who become majors who don't understand the nature of the discipline.

These issues have been attacked on many fronts, but the approach of interest in this research is that of incorporating simple robotics into the pre-curriculum and more advanced robotics for undergraduate research and development.

Robotics captures the imagination, both as a means of accomplishing tasks, and as a model of human behavior. The area encompasses both the electro-mechanical side of creativity, and the software/behavioral side. We have used robotic technology to stimulate interest of potential students through incorporating robotic demonstrations and hands-on experience. We have also used robotics with current students, first as an approach to teaching logical problem solving, and then as a tool for advanced research and development. These different uses of robotics complement each other, thereby forming one method of addressing the problems of recruiting and retention.

2 Problem Formulation

Montana Tech is primarily an engineering school, and approximately half of our student population comes from the state of Montana, although some of our programs draw national and international students. The computer science department generally draws the majority of its students from in-state. Montana is a sparsely populated state, with most areas considered rural. Because of this, computer science, and rigorous science beyond the basics, is not emphasized in most pre-college curricula. Consequently many students entering the university are not aware of computer science as a discipline or a career path.

In addition to lack of awareness of the field, many high schools don't prepare college-bound students for the more rigorous mathematical requirements of engineering and science required at the university level. Many students entering our program are underprepared mathematically. A significant number of students declare computer

ISBN: 978-1-61804-085-5

science as a major based on their enjoyment in using computers either for schoolwork or for gaming. Unfortunately, neither of these uses of computers truly shows the logical and mathematical nature of the discipline. Many students who enroll in the introductory course sequence become disillusioned early when they discover the exacting nature of programming.

In this paper, we discuss the problem areas we addressed by incorporating robotics in this section, then the approaches used to address those problems in Section 3. Finally, we talk about work under development and future directions in Section 4.

2.1 Computer Science Awareness

During the 2008-2009 academic year, as part of its investigation into recruiting efforts, our department hired consultants to conduct focus group studies of local high school attendees. We were particularly interested in finding out whether the cultural perception of computer science majors by young women was "geeky", and if that was the reason we were attracting few women into the program. The results of the focus group work were eye-opening. It was discovered that young women in local high schools had not even heard of computer science as a potential career path, and therefore had not even considered it as a major.

Additional contact with high schools outside of our local area uncovered that very few schools included any kind of computer science in their curricula, and those that did had very few students enrolled. In one of the more populous cities in the state, three of the high schools (jointly comprising a population of several thousand students) joined together to offer a single course in computer science. The class had an enrollment of 7 students. This is not an indictment of high schools – with education funding tight, the schools must focus on the core subjects: English, mathematics, and basic sciences. Computer science was a subject that very few could afford.

This revelation posed a challenge in our recruiting efforts. If students are unaware of computer science as a potential path of study, how could we make our program attractive to them?

McKee and Maunders found an interesting approach to allowing the public to interact with robotics using the internet [1]. They used a customized environment in which students developed a web-interactive application where a user controls a toy "digger" to find, move to, and pick up a ball within the environment. Their work supports one of the approaches we are taking to making robotics control available online as an approach to raising awareness of the computer science discipline in general, although we are approaching it using the Aldebaran NAO H25 robot and off-the-shelf technology.

2.2 Recruiting

Assuming we could make more potential students aware of computer science as a major, the next question becomes "how can we attract them into *our* program?" What about our program makes the curriculum more engaging than another computer science program? And how can we spread the word about our program?

2.3 Engaging and Educating

Many of the students who enter our program are mathematically underprepared for the computer science curriculum. In fact, this is not unique to the computer science program, but applies to other engineering disciplines as well. Montana Tech has addressed this problem on a campus-wide scale by designing the Fundamentals of Engineering and Science Program (FESP). FESP is designed to bring the math and science skills up to a college entry level for all students who are under-prepared. At its inception, however, the program focused on mainstream engineering students, and didn't meet the needs of our computer science and software engineering students.

The FESP students who wished to enter computer science or software engineering programs don't meet the prerequisites to enter our standard degree track program, and in the past have been routed into programming courses that don't count toward their degree, but ones that would hopefully give them a taste of computer science; courses such as Applications Programming (Visual Basic) or Matlab Programming. While these courses do teach programming, they don't teach the more disciplined and mathematical science of computation.

The lack of preparation, and potentially the lack of understanding of computer science as a discipline led us to ask "how could we engage these students, teach them the basics of logical thinking, and introduce them to programming at the same time?"

In younger students (ages 9-15), Pucher and Hofmann found that the "one-way, top-down transmission of knowledge" is not as effective as a more hands-on project-based approach, particularly in stimulating and motivating learners to solve real-world problems [2]. In their research, the Lego Mindstorm NXT and its graphical programming environment were used to engage students in learning and problem solving. Although our

ISBN: 978-1-61804-085-5

students are older once they enter the university system, we believe that some of the same approaches can be used to engage them.

Hsu and Weng also used the Lego Mindstorm NXT robots to teach robotic control through programming using the Java language [3]. They found that first introducing programming through the graphical programming environment and then through annotated Java code enabled students to understand the programming process more easily. Again, we feel that this approach is more applicable to our entry students who are under-prepared for the computer science curriculum.

2.4 Advanced Education and Research: Addressing Retention

Once students have fully entered our program, the next question becomes "how do we continue to engage students, and provide a state of the art education that will be applicable to either graduate study or the career market?" That is, how do we ensure that we retain those students who are prepared and interested in a career in computer science?

Ray et.al. discuss engaging online learners by considering learning style [4]. They discuss that learning styles of computer students tends to be active, sensing, visual, and sequential, as opposed to reflective, intuitive, verbal and global, and they use these findings in structuring their online computer science course to engage students. We do not have an online program, but the fact remains that computer science students are more engaged when we approach teaching by considering appropriate learning styles.

Jehlicka discusses the interdisciplinary nature of programming [5], and used Lego Mindstorm NXT robots to illustrate and teach students about physical phenomena such as the light spectrum and light combination. While we use the robotics to teach programming skills, it is a nice side-effect that our students are learning about the physics of robotic motion and control also.

3 Problem Solution

The introduction of robotics into our department curricula was not the initial approach tried, nor is it the only method we are using to address the issues discussed. It is, however, one of the more visible aspects of our approach, and the literature has shown that robotics is an effective approach to stimulating, motivating and engaging learners, a result we wish to achieve in our program.

In each of the identified problem areas, we have incorporated robotics technology in different ways, with different intents. We discuss each of our uses of robotics in the following sections.

3.1 Raising Computer Science Awareness with Robotics

In order to increase the awareness of computer science to potential students, we have incorporated the use of robotics both as demonstration tools and for hands-on experience.

4-H Club is an international organization for students ages 9-19, and its mission is to provide leadership, citizenship and life skills to its members. Local grade school and junior high school 4-H members started using our Lego Mindstorm NXT robots last year. The department provides mentorship to assist these students with their projects through our own computer science students.

The department has also sponsored workshops for high school students using the Mindstorm robots. In this case, a project was devised in which students were to build one of three types of robots, program their robot using the graphical programming environment available with the kits, and then have their robots play a competitive game.

The Mindstorm robots are also used as demonstration tools when potential students visit our department.

Finally, with the acquisition of the Aldebaran NAO H25 robot, we are developing a way for potential students to interact with and control the robot via the internet. This project is discussed more fully in section 3.4

Since we are working with younger age learners, results of our efforts will not be known until these potential students reach university age.

3.2 Robotics for Recruiting

Many of the same strategies for raising computer science awareness are expected to act as a venue for recruiting also. Again, results of our efforts may not be conclusive for a few years, when learners enter the university system.

3.3 Robotics for Engaging and Educating

In working with students who are not prepared for the computer science curriculum, we added a class that would be targeted toward those who had an interest in learning programming.

Following the lead of Jeannette Wing of Carnegie Mellon University (CMU) [6], we introduced a course called Computational Thinking designed to teach problem solving skills to underprepared students entering the curriculum. Diverging from the CMU approach, we also incorporated simple robotics. The course is taught as two hours of lecture and three hours of lab per week. The lecture portion stresses logical and mathematical thinking and problem solving. The lab portion uses the Lego Mindstorm NXT robotics kits to illustrate some of the concepts taught during lecture. The robotics approach was chosen rather than general programming because of the interactive and very visual demonstration of results that students get. Students in the lab must build their own robots and program them, first using the graphical programming environment provided with the kits, and then moving on to programming them in Java (using LeJos, a Java library for the NXT robots). We have offered this course for three years now, and have been able to gather concrete measurements on its success.

To test the effectiveness of incorporating the Computational Thinking course and robotics into the pre-curriculum, the Whimbey Analytical Skills Inventory (WASI) pre-test and post-test were used in each of the three years of offering the course [7]. On the first day of class, the WASI pre-test was administered. It consists of 37 problems that test mathematical and logical thinking in various forms. During the semester, instruction is given which first addresses the types of problems the WASI test covers, and then delves into deeper logical concepts such as algorithm development, using structured logic, invariants, and recursion. Students are not informed they will be taking the WASI post-test; in fact, they are told that the final exam will cover class material. On the last day of classes, the post-WASI was administered. This is a second version of the WASI test, also consisting of 37 problems of similar composition as the pre-test.

Comparison of WASI pre-test and post-tests was done over the three years of administering the course. In the results, scores were included only if a student took both of the tests. (Not all students are present for the first day of classes, nor do all students attend class, particularly if they are unaware that an exam will be administered.) An additional pair of scores was discarded because the student misunderstood the instructions and answered none of the questions on the pre-test, resulting in a 33 point gain between pre- and post-test scores. Including this data point would have skewed the data quite favorably, but would be an inaccurate measure of the true success.

Initial (pre-WASI) test scores ranged from a low of 12 to a high of 28, while post-test scores ranged from a low of 19 to a high of 37. The average point improvement over the course of a semester was 6.92. Running the student's t-test (single-tailed matched), the improvement between pre-and post-test scores turned out to be highly significant, with $p<0.01$ ($p=0.0000267$).

The significant difference between performance on the WASI pre- and post-tests indicates that the course, with its use of robotics, impacts the problem solving ability of these students. Over time, we intend to track the effectiveness of the course long-term, that is, to see how students who took the course perform in our courses as opposed to those who were similarly under-prepared, but didn't take the course.

3.4 Robotics for Advanced Education and Research

Once students understand computer science as a discipline, and they have entered our degree track program, we wish to continue to engage them and make their learning experience as relevant as possible. To this end, we have used the Lego Mindstorm NXT robots in some course work and as independent research tools. However, the capabilities of the Mindstorm robots reach a limit. To this end, we purchased a NAO H25 robot for student research use. We hope to eventually grow our robotic capability to be able to enter the RoboCup competition as has [2].

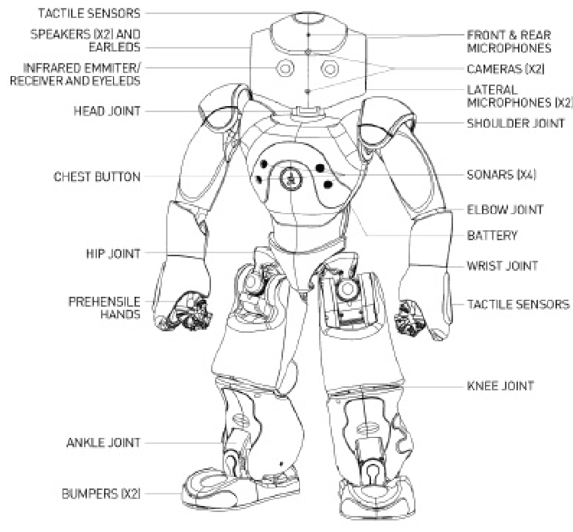

Figure 1: NAO H25 Robot

NAO is a humanoid robot manufactured by Aldebaran Robotics. NAO is roughly 58 cm tall, has 25 degrees of freedom, a variety of sensors (vision, tactile, auditory, sonar, etc.), and is capable of network connectivity via Bluetooth, 802.11, and infrared [8], [9]; all of which make NAO an

ISBN: 978-1-61804-085-5

extremely versatile robot, and capable of engaging students at a much more technical level than the Mindstorm robots.

NAO H25 is the standard humanoid platform for the RoboCup competition, and has been used in many other areas, including voice recognition, audio and visual signal treatment, and trajectory optimization. Aldebaran Robotics has been encouraging NAO development in academic settings as well. Their premise is, which agrees with the literature, and complements our approach, is that robotics can be an engaging approach to stimulate student interest in computer science and similar fields of study.

NAO provides a platform for both beginners and advanced users. Choreographe is a graphical programming environment that allows users to control the robot with little programming experience. Additionally, Aldebaran includes a software bundle and documentation for their "NaoQi" middleware pre-installed on NAO. NAO may be programmed with a wide variety of languages including C++, C, Python, Java, and Urbi.

Figure 2: NAO and NXT "Dog": The dog has a touch sensor that is depressed if the harness pulls, providing a signal to the dog to stop moving.

3.4.1 NAOCam

The NAO robot is a fairly large investment, and it is unlikely that potential students would have access to such technology. The idea for the NAOCam project came about as an opportunity to allow remote programming and control of NAO via the web. The project therefore contributes to our goal of increasing awareness to the public about computer science, while at the same time engaging our upper level undergraduates in developing the NAOCam system.

The design of the system is as follows. The server host will set up a web service. The server host will need to have software to allow a connection from the web server to NAO. With connectivity, then, a remote client is able to control NAO from anywhere on the Internet, given the web server information.

Since operation will be remote, visual feedback to the client is necessary. This will be accomplished using a webcam feed. The remote client can subscribe to the webcam feed, connect to the web server, and begin testing either existing code on the server, or their own code.

The web server will have the Aldebaran software available for clients to use, including NaoQi and the SDK for Python or C++. Python, in particular, is a language that has been used in teaching early programming.

The system is so far in the design stage, and there are two issues that still need to be resolved. Since there is only one physical NAO robot, scalability of this system is an problem. Only one script/program can be run at a time. However, the webcam feed will allow as many viewers as the web server permits, so multiple clients can still observe NAO simultaneously. Depending on usage, it may be necessary to allocate time slots to remote clients. Ideally, additional NAO robots could be purchased.

The second issue with the system is the security and safety of the NAO robot. It is not guaranteed that the remote client will program or control NAO in a safe manner. One possible solution is that NAOCam would be made available with pre-programmed code blocks for users to select and combine, but which have been tested for safety with NAO.

4 Conclusions and Future Work

Robotics is being used in the Montana Tech Computer Science Department in a variety of ways: to increase awareness of the field, to assist us in recruiting students, to engage our existing students, and to provide an advanced research and development experience for our upper level students. Some of our efforts have shown significant effect, particularly with the use of robotics in our Computational Thinking course. Other efforts will take time before results are expected. And finally, our NAOCam project is still in its design phase. Thus far, though, our results appear promising, and we are optimistic that as additional work is completed, results will also be positive.

ISBN: 978-1-61804-085-5

There are several directions into which we wish to take the robotics work. The first is that of fully developing the NAOCam system for public access. A second direction is that of working with the Montana Tech system to make the Computational Thinking course count toward general education requirements. This would provide two benefits. One is that more students (not necessarily only computer science students) would be willing to participate in the class if it counted toward graduation, and thus more students would gain awareness of computer science as a discipline. Another benefit is that our own majors who are not yet prepared for the mathematics in the computer science curriculum will have a course that teaches the science of the discipline, and not just the programming piece. Finally, we intend to program and use the NAO robot as a recruiting tool along with the Mindstorm robots. To this end we would like to make NAO as interactive and human-friendly as possible.

References:

[1] G. McKee and R. Maunders, “Exploiting Toys and the Internet for Robotics Education”, *Proceedings of the 2001 WSEAS/IEEE MultiConference on Modern Information Technologies and Robotics*, 2001, Malta.

[2] R. Pucher and A. Hofmann, “Teaching with Projects Increases the Intrinsic Motivation to Learn – The RoboCup Project”, *Proceedings of the 10th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics (CIMMACS '11)*, 2011, Jakarta, Indonesia, pp. 179-182.

[3] M. Hsu and T.-S. Weng, “Assisted Instruction Case Design of Robot Creative Assembly and Control Program Design”, *Proceedings of the 7th WSEAS/IASME International Conference on Educational Technologies (EDUTE '11)*, 2011, Iasi, Romania, pp. 106-111.

[4] S. Ray, A. Denton, C. Beseman, and K. Nygard, “Learning Theory and Styles in Online Computer Science Courses”, *2004 WSEAS Workshop: Engineering Education*, Venice, Italy, November 2004.

[5] V. Jehlicka, “Interdisciplinary relations in teaching of programming”, *Proceedings of the 2010 WSEAS Conference on Applied Computing*, 2010, Timosaora, Romania, pp. 33-38.

[6] J.M. Wing, “Computational Thinking and Thinking About Computing,” *Philosophical Transactions of the Royal Society*, vol. 366, July 2008, pp. 3717-3725.

[7] A. Whimbey and J. Lochhead, *Problem Solving and Comprehension*, 6th ed., Psychology Press, Taylor and Francis Group, New York and London, 1999.

[8] Aldebaran Robotics, http://www.aldebaran-robotics.com/en/Pressroom/About/NAO.html, 2012

[9] Aldebaran Robotics, http://developer.aldebaran-robotics.com/nao/, 2012.

[10] R. Ramli, M. Yunus, and N. Ishak, “Experience of Robotic Teaching for Malaysian Gifted Enrichment Program at PERMAT Apintar”, *Proceedings of the 9th WSEAS International Conference on Education and Educational Technology (EDU '10)*, 2010, Iwate, Japan, pp. 163-166.