Iterative Design of a Robot-Centered Curriculum for Introduction to Computer Science

Tom Lauwers and Illah Nourbakhsh

Carnegie Mellon University
T. Lauwers, Newell-Simon A504, Carnegie Mellon University, Pittsburgh PA
I. Nourbakhsh, Newell-Simon 3115, Carnegie Mellon University, Pittsburgh PA
tlauwers@andrew.cmu.edu, illah@cs.cmu.edu

Introduction

In the context of steeply declining enrollments in Computer Science [6], our group is focusing on developing curricular modules for introduction to Computer Science (CS1) classes in which robots are used as educational tools to motivate students about applications of Computing.

We strongly believe that robots can be an ingredient in the solution to the retention and diversity problems plaguing Computer Science education. Robots have been used in a number of contexts to excite students towards further study in STEM (Science, Technology, Engineering and Mathematics) fields. At Carnegie Mellon, a student taught course in which programming robot behaviors and creating robotic art are the main activities attracts nearly half its students from the humanities, business, and art schools [5]. High school robotics competitions such as FIRST and Botball have grown explosively over the last decade. Robot kits, such as Lego's Mindstorms, have also become a popular tool in traditional CS1 [1] and CS2 [3] courses.

Robots, as physically manifested computing devices, inherently show students how computing algorithms can impact the real world; they provide a degree of relevance to assignments that is often missing. Even so, there are well-known weaknesses to using robots in computing courses [1]: Robots are typically too expensive for student ownership, and so students must work on robot programming assignments in labs with limited hours. Feedback is delayed due to the real-time nature of robotics, and so students must devote more time to tedious debugging, and less to developing solutions. The ingredients needed to overcome these limitations are achievable: New hardware technologies [4] that enable high functioning robotics at a low cost; a software architecture that is student-friendly and can be peeled away to reveal layers of deepening complexity as students learn and become more sophisticated users; and a simulator to provide immediate debugging feedback.

Approach

We submit that the reason that robotics-basic curricula have not succeeded in CS1 classes is a lack of alignment between the features and capabilities of the robotic technology and the needs of the curriculum and students. Our goal is to develop a curricular framework with an accompanying robot whose features have been selected specifically to support the needs of the class, students, and curriculum.

To meet this goal, we are engaging in an iterative design process with extensive participation from computer science educators at two and four year schools. The process is composed of the following steps:

Design. The (re)design step involves the creation of a robotic platform and associated software, and the development of assignments and course outline.
Pilot. The pilot phase involves the pilot of the designed curriculum and technology in a CS1 course.
Evaluation. Learning, motivation, and retention are tracked with standard exams, weekly student surveys, and comparisons of drop-out rates. These data are used to inform the next design phase.

Initial Activities

Prior to starting our first design phase, we conducted a survey of educators teaching CS1 at two and four year school institutions [ref]. This survey had a number of purposes; it acted as an initial evaluation step for standard CS1 classes, it provided an understanding of the typical challenges facing CS1 educators, and it suggested to us that using robotics in CS1 was an idea which was both acceptable to many educators and was a change that could be made without extensive administrative oversight.

Armed with the results of our survey, we started the design process by recruiting educators to work with us; we are currently working with two educators teaching CS1 at community colleges. These educators have become true design partners, suggesting changes to software and hardware, coming up with assignment ideas, and affording us opportunities to pilot our curriculum.

Curriculum design was guided by several conclusions from our survey; educators teaching Java were most enthusiastic about using robots, most educators wanted a new curriculum to be linked to a textbook, robots could be introduced into a class with minimal administrative paperwork so long as the programming language did not change and the course schedule was minimally affected. We focused on Java and chose a specific textbook to link to our curriculum. We took care to ensure that each assignment developed was focused on teaching CS concepts; we used the prior assignments from our design partners as a baseline and analyzed these to ensure our assignments were similar conceptually. We also minimized the amount of robot-specific information students need to know.

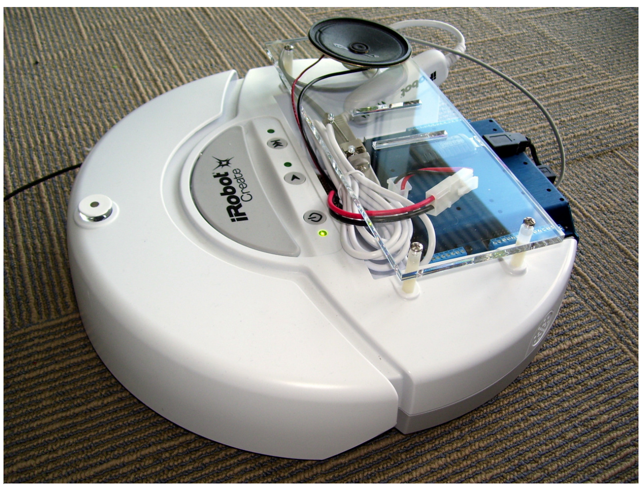

Figure 1: First Design of Robotic Platform

Our technology goal for the first design cycle was to develop a robotic platform which had a large set of features at a reasonable cost; our aim was to maximize the number of features which could be tested, and then develop an optimal feature set to guide our second design. Our final design (figure 1) is an iRobot Create robotic platform combined with a Qwerk controller; the Qwerk is wirelessly tethered to a student's computer. This combination of technologies has the following capabilities: bumper-based obstacle detection, encoder-based movement tracking, vision through a webcam, full audio including arbitrary text to speech, color and intensity variable LEDs, and access to real-time RSS feeds.

In Fall 2007 we began two pilots with our design partners (figure 2). These pilots involve over 60 students and 15 robots at two community colleges, one in Pennsylvania and one in California. In the Pennsylvania course, our design partner has kept detailed records of retention and performance of students going back six years; our course will track performance and retention, and compare them with previous figures as well as the

Figure 2: Students working in the robotics lab

retention rates of a comparable course taught concurrently by another educator. Additionally, we are surveying students upon the completion of each assignment to determine which assignments were interesting, and to track motivation throughout the semester.

We will use the data from the Fall 2007 courses to determine which assignment themes and technological features to keep and expand upon and which to remove. Once this analysis is complete, we plan to redesign both the curriculum and technology in Spring 2008, and pilot the new course at three or four educational institutions in the Fall of 2008.

References

[1] Fagin, B. & Merkle, L., "Measuring the Effectiveness of Robots in Teaching Computer Science." *pg. 307-311, SIGCSE 2003.*

[2] Lauwers, T. & Nourbakhsh, I., "Informing Curricular Design by Surveying CS1 Educators", *In Proceedings of AMiRE, 2007.*

[3] McNally, M., "Walking the Grid: Robotics in CS2." *In Proceedings of the 8th Austalian conference on Computing education, pg 151-155, 2006.*

[4] Nourbakhsh, I., et al., "A Roadmap for Technology Literacy and a Vehicle for Getting There:Educational Robotics and the TeRK Project." *In Proceedings of IEEE RO-MAN 2006.*

[5] Shamlian S. et al., "Fun with Robots: A Student-Taught Undergraduate Robotics Course." *pg. 369-374, ICRA 2006.*

[6] Vegso, Jay. "Interest in CS as a major drops among incoming freshmen." *Computing Research News, May 2005.*