Informing Curricular Design by Surveying CS1 Educators

Tom Lauwers[1] and Illah Nourbakhsh[2]

[1] Carnegie Mellon, Pittsburgh, PA, USA, **tlauwers@andrew.cmu.edu**

[2] Carnegie Mellon, Pittsburgh, PA, USA, **illah@cs.cmu.edu**

1 Background

In the context of steeply declining enrollments in US Computer Science education [7], our group is 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. The CS1 class is traditionally taught in the United States as an in-depth introduction to the art of programming from primitive data types through control structures and objects, ending with arrays and exceptions. It is not designed as a broad-based introduction to Computing; students practice computing concepts through implementation in a real-world programming language.

We strongly believe that robots can be an ingredient in the solution to the retention and diversity problems plaguing CS education. Robots have been used in a number of contexts to excite students towards further study in STEM (Science, Technology, Engineering and Mathematics) fields. Robots have become popular tools in traditional CS1 [1] and CS2 [4] courses, with new technologies [5] enabling high functioning robotics at low cost. 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 [6].

To ensure that our curricular designs are grounded in current CS1 classroom realities, we embarked on an extensive survey of educators at two and four year institutions. We were inspired by the Taulbee survey [8] and the McCauley and Manaris [3] studies, but our aims differed from these – instead of a broad based, largely quantitative analysis of the state of Computer Science education, we were interested in analyzing the attitudes, opinions, and challenges faced specifically by CS1 educators. Our study sought to answer questions that were best asked in the context of a personal interview, and best analyzed through conceptual code-based qualitative metrics. The foundational questions which we sought to answer included:

- How do CS1 instructors feel about the effectiveness of their curricula, both in teaching students and in motivating them?
- To what degree are instructors able to make curricular changes?
- What are the typical dynamics and logistics of a CS1 course?
- What tools and programming languages do instructors currently use and what is their relative popularity?
- Are instructors interested in using robotics as a teaching tool?
- How do the classroom realities exposed by the previous questions inform methods for introducing new educational tools?

2 Participants

120 educators were identified as currently teaching CS1 at four year institutions within the United States. Of these 120, 33 responded to an email request to participate in the survey. These participants and the institutions they represented were diverse with respect to gender, professional level, public/private status of the university, and size of the university. A follow-on study was conducted with four educators at Pennsylvania community colleges to determine if these educators' opinions diverged from those of the four year group. Educational contexts differ significantly between two and four year institutions: American community colleges offer two-year degrees and focus on skills and workforce training. In many cases students spend one or two years at the more affordable community college and then transfer to a university as a second or third year student.

3 Analysis

The phone interview was composed of three main sections; course logistics, ability to modify the CS1 course/interest in doing so, and knowledge of and interest in robotics. The analysis of the interviews was performed in three parts:

- A qualitative analysis was performed on open-ended questions which sought to examine the frequency with which a concept was expressed in response to a given question.
- Basic statistical methods were used to find the mean, median, and standard deviation of questions which resulted in quantitative answers: Five questions asked for interviewees to place their opinions on a 1-5 scale, while another concerned the total number of hours students should work on the class.
- A correlational analysis was performed to determine if the expression of certain opinions or characteristics predisposed an educator to other opinions or characteristics.

4 Results From Four Year Institutions

The survey has provided us with a rich source of data to inform our curricular design work. Only results which speak directly to the foundational questions asked in the opening of this paper are presented. A technical report with results for all questions and supporting materials [2] has been published to allow a more extensive reading and review. Results fall into several categories; attitudes of CS1 educators about their curriculum; ability to change the curriculum; results concerning class logistics; programming languages and educational tools; and attitudes specific to the use of robotics.

4.1 Satisfaction and Empowerment

Educators were asked to rate on a 1-5 scale how excited they felt their students were by the class, and how satisfied they themselves were with their curriculum. The mean answer for student excitement was 3.24, while for teacher satisfaction the mean was 4.03. More than 80% of the educators rated the level of satisfaction with their CS1 curriculum as 4 or better. These two ratings were correlated to investigate whether a higher satisfaction with the curriculum corresponded to increased perceived student excitement. No such correlation existed; the satisfaction a teacher feels with their CS1 curriculum is *not* correlated with perception of student excitement.

In order to determine how empowered individual instructors were by their respective institutions to make changes to their CS1 curricula, we asked what the administrative process was for initiating major changes, such as changing the programming language, and minor changes, such as a change in the class schedule. Only four instructors were able to make a major change with a low level of administrative oversight. The other 28 respondents listed a variety of processes, all requiring approval from either all other CS instructors, a departmental curriculum committee, and/or an interdepartmental committee composed of faculty from all affected departments. This is not to say that CS1 educators are not able to make changes to their courses – 22 respondents were able to initiate minor changes to their course with no oversight, and an additional seven were able to do so with approval from their peer introductory course instructors.

4.2 Time Commitment and Cost of Materials

The total weekly amount of time students are expected to spend on a CS1 class varies widely, but fits a normal distribution with a mean and median at 10 hours per week and a standard deviation of 3.14 hours. In class time was fairly homogeneous with a minimum of three hours, a mean of 4.39 hours and maximum of six hours per week. Time spent out of class was much more varied, with a minimum of one hour, a mean of 5.89 hours and a maximum of 14 hours. The location of out of class work was almost always expected to be in a general computer lab or on the student's personal computer; only three participants required their students to complete assignments in a computer lab with pre-set hours (a closed lab).

Educators were asked to rate the importance of material costs such as textbooks and software for their students and for their departments on a 1-5 scale. With regards to the cost to the student, most educators are very concerned with student costs, with more than half rating a 4 or higher on the 5 point scale. The answers to the question about departmental costs were highly varied, representing large differences in the budgets of different computer science departments.

4.3 Programming Languages and Teaching Materials

Of all respondents in the survey, 17 used Java, 11 used C++, 3 used C, 1 used Scheme, and 1 used Visual BASIC. In addition to determining the programming language, we also asked educators whether they had changed the programming language or environment in the last two years, and if they were planning to change language and environment in the next two years. Nineteen educators did not change **and** are not planning to change either the language or environment. Of those who made changes, three educators have recently switched to Java, and one has recently switched to C++. In the next two years, no Java instructors plan to switch languages and one C++ instructor is planning to do so. Java instructors recently and in the near future are more likely to switch programming environments, possibly due to the proliferation of Java IDEs.

Instructors were asked about the divergence between their curriculum and the ordering of lessons in their chosen textbook. Eleven instructors diverged either a little or not at all from the text, eleven used the text as a resource, assigning readings and problems, but choosing in which order the topics were covered, and eleven did not have a main text for the class, relying instead on instructor notes.

In a separate question, instructors imagined themselves to be planning a new CS course at the introductory level, and were asked about the support materials they would need in order to do so. 23 of the 32 responses expressed a need for a main textbook to use for assignments, examples, and to provide a basic course roadmap. Interestingly, instructors who do not currently use a text were just as likely to desire one when placed in this hypothetical situation as those instructors who rigidly stick to the textbook.

4.4 Robotics

We queried instructors' views about the effectiveness of using robotics as a teaching tool in their course. Of the 31 responses to the question "Can you see ways in which robotics could augment your current introductory CS course?" nine made wholly positive comments about the notion, 20 pointed out both positive and negative aspects of using robotics in their classes, and two made exclusively negative comments. Robotics was not mentioned in the recruitment email, so we do not believe that there was a response bias. The most common positive comments regarding robotics were that it would improve student motivation and increase learning by concretely high-lighting concepts. By far the most common negative comment about using robotics was that it was too time-consuming. Other negative comments about robotics were that it was expensive, made grading more difficult, constrained the programming language, and that hardware debugging might distract from learning CS.

We also correlated the educator's views on robotics with other characteristics. At schools at which robotics courses were offered, instructors were more positive about using robots, and the more familiar they were with the courses currently offered, the more positively they responded. Our most surprising correlation was found between the programming language used in the CS1 class and interest in using robotics. The instructors using Scheme and Visual BASIC were both wholly positive about using robotics, as were 7 of the 15 Java instructors. By contract, instructors teaching C/C++ were significantly more negative; the difference in attitudes between those educators teaching C/C++ and those teaching Java was statistically significant despite the small sample. (p-value of 0.014)

5 Results From Two Year Institutions

For the most part, the responses of the community college instructors mirrored those of their counterparts at four year schools. Although not statistically significant due to the extremely small sample, there were some notable and interesting divergences:

- The average expected time spent on the class was much higher (14.75 hours vs. 10 hours per week) at Community Colleges.
- While educators rated student excitement at a similar level (3.38 vs 3.24), self-reported satisfaction was lower (2.88 vs 4.03).
- Community college educators rated the price of materials to the department as very unimportant. Student costs were rated as very important by all four community college educators.
- All of the instructors stuck closely with their chosen textbooks – three said the syllabus mirrored the textbook exactly, one said that it diverged slightly.

6 Discussion

Our results have provided tentative answers to the questions posed at the beginning of this paper and feed into answers to our primary question – what methods and strategies should we employ in our attempt to introduce a new educational tool into the CS1 curriculum? Our conclusions can be split into those general to introducing a new educational tool and specific to introducing robots. We begin with general conclusions:

- A major curricular change is difficult to implement, requiring buy-in not just from the instructor, but from the department faculty, and sometimes from other departments. Tools should therefore be introduced with a curriculum that does not deviate from high-level concepts currently covered in the typical computer science course. To minimize impact on the existing curriculum, tools can be introduced through self-contained curricular modules that cover a major course concept.
- Tools should not be tied to a new or relatively rare programming language; not only does this require a major curricular change but few instructors are planning to change the programming language they use. Support for either Java or C++ is required for widespread adoption.
- Any new educational tool must integrate into the class such that it does not significantly increase student workloads.
- Given that most educators feel textbooks are important when embarking on a significantly different curriculum, it may be useful to either develop an accompanying textbook for the educational tool, or choose a popular existing textbook and develop a modified lab manual to accompany the book. This is especially important if the tool will be deployed at community colleges as well as four year schools.

Specific lessons relating to using robots in CS1 are:

- Students must be able to work on their out of class assignments at home. This is not an impossible goal for a robotic tool – we intend to meet this challenge through the use of a simulator. Depending on the cost of the robot, it may also be possible in some venues to allow students to take the robots home.

- We are initially developing for Java, providing us access to the largest and most enthusiastic base of potential robotics adopters.
- Materials cost to students is widely perceived as more important than departmental costs, and so we do not believe that a robotics class should require students to purchase a robot unless it is in lieu of a similarly priced textbook.

Our survey has provided us with a foundational experience base from which to launch our curriculum design efforts. We hope that this research can be of similar use to other curriculum and technology designers interested in improving CS1.

References

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

[2] Lauwers, T. and Nourbakhsh, I. (2007) Supplementary Materials and Results of A Survey of CS1 Educators. Technical Report, Carnegie Mellon University, Pittsburgh, PA

[3] McCauley, R. and Manaris, B. (2002) Computer science education at the start of the 21st century - a survey of accredited programs. In 32nd ASEE/IEEE Frontiers in Education Conference

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

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

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

[7] Vegso, Jay (2005) Interest in CS as a major drops among incoming freshmen. Computing Research News

[8] Zweben, Stuart (2005) 2003-2004 taulbee survey: Record ph.d. production on the horizon; undergraduate enrollments continue in decline. Computing Research News