Computer Science Education

ISSN: 0899-3408 (Print) 1744-5175 (Online) Journal homepage: http://www.tandfonline.com/loi/ncse20

Robotics in computer science education

Jennifer S. Kay & Tom Lauwers

To cite this article: Jennifer S. Kay & Tom Lauwers (2013) Robotics in computer science education, Computer Science Education, 23:4, 291-295, DOI: 10.1080/08993408.2013.856614

To link to this article: http://dx.doi.org/10.1080/08993408.2013.856614

Published online: 14 Nov 2013.

Computer Science Education, 2013
Vol. 23, No. 4, 291–295, http://dx.doi.org/10.1080/08993408.2013.856614

Routledge
Taylor & Francis Group

EDITORIAL

Robotics in computer science education

Robots would not be robots without computer science. Leave out computer science and what remains are fancy mechanisms and remotely controlled machines. It is no surprise then that robots appear in numerous computer science courses. The papers in this special issue, like the title of the issue itself, represent two distinct topics. The first two papers investigate the use of robots in teaching computer science concepts to a general audience. The final three papers study how to best teach concepts in robotics to upper-level computer science students. Taken together, these papers show the use of robotics in education with 11-year-old school children all the way up to students pursuing graduate studies in computer science.

Robots have been used for education since 1971 (Feurzeg, 2006) when the Logo "floor turtle" (Papert, 1980) was introduced to teach logic and computing concepts to children. Using Logo, children could program the hemispherical floor robot "turtle," to move a specified distance, turn, beep, and draw shapes on paper using a small pen placed under the turtle's body. Roughly, a year later, the same team introduced the "screen turtle" simulator (Feurzeg, 2006) that was widely used by children in the 1980s.

The 1990s saw several major developments that accelerated the use of robots in education: the creation of the first low-cost microcontrollers, such as the BASIC Stamp (Edwards, 1998) and the Handy Board (Martin, 2001), provided students with the ability to create robots around programmable devices. The founding and explosive growth of robotics competitions, notably FIRST (http://www.usfirst.org/) and Botball (http://botball.org/), raised awareness of robotics among high-school teachers. Finally, the development and mass-market sale of a reconfigurable robotics kit, the Lego RCX (Cliburn, 2006), allowed even elementary school students to build and program a robot in just a few hours.

By 2000, computer science educators at the college level were beginning to explore using robots as tools in introductory computer science classes. Early results were not always promising. In an impressive study, Fagin and Merkle (2002) evaluated the performance of 938 students taking a core computing course required of all students at the US Air Force Academy. Roughly, one-quarter of these students were in special robotics sections of the class in which students used Lego Mindstorms Robots in four out of the

six laboratory exercises. Much to the dismay of the robotics community, test scores were lower in the robotics sections of the course than in the traditional sections, and the use of robots did not affect students' choice of discipline. Fagin and Merkle surmised that logistical difficulties (limited student access to robots, long delays between compiling and debugging a program) were the source of the poor performance of these students. This hypothesis is consistent with research in traditional classroom manipulatives that emphasizes the importance of students having sufficient hands-on time with the equipment (Huetinck & Munshin, 2004).

The price of robot hardware has continued to drop, and it is now possible to teach a class in which every student has their own "personal" robot. When Summet et al. (2009) compared a CS1 course using robots (in which each student had a personal robot they could use whenever they wished) with a non-robot course, their initial results showed that the robot group performed better on several common exam questions. However, they could not conclude with certainty that robots improved learning over non-robot equivalent classes. While we cannot point to definitive proof of the benefits of educational robotics, there are lots of positive signs: we ourselves (Kay, 2010; Lauwers, 2010) have found indications that the use of robots increases student enthusiasm and engagement.

The five papers in this issue were selected from among 14 full papers submitted for review. Each paper was reviewed by three experts who were not affiliated with the paper authors. The reviewers rated the selected papers significantly higher, both quantitatively and qualitatively, than those papers that were not included. We both hold degrees from Carnegie Mellon and are cognizant of the fact that three of the papers in this issue include at least one author with Carnegie Mellon affiliations. Three of our 14 reviewers are currently affiliated with Carnegie Mellon and two additional reviewers have past affiliations. However, none of the papers in this issue that have Carnegie Mellon authors were reviewed by individuals with Carnegie Mellon affiliations.

The first two papers are excellent examples of how well-designed studies can produce significant results even with relatively small sample sizes. Mason and Cooper discuss a series of experiments in which school children were introduced to programming using robots in short hands-on workshops. While the age, gender, and selection criteria for the workshops vary, there is an impressive unifying result: a statistically significant increase in confidence in programming. Beyond this, each individual study showed statistically significant improvements in other areas, such as a reduction in the perceived difficulty of programming and an increase in participants' intent to consider a career in CS/IT. Perhaps the most interesting result, however, was the effect that the complexity of the robot programming interface had on participants' subsequent learning in a non-robot programming environment. Mason and Cooper also detail how Cognitive Load Theory guided their workshop design and supports this final result.

Liu, Schunn, Flot, and Shoop investigate the introduction of the Robot Virtual Worlds (RVW) simulation environment in several high schools. RVW allows students to use the same code to program real and simulated robots, allowing these authors to investigate the utility of a physical vs. simulated robot in the context of a high school class. The results are at times contradictory and subtle, and worth a detailed read. To summarize, students were able to progress through the curriculum significantly faster when using a simulated robot only, with no impact on learning. However, in another study detailed in the paper, students learned algorithmic thinking better when programming a real robot.

Continuing the theme of simulation, Moll, Bordeaux, and Kavraki present the Open Motion Planning Library (OMPL), a ROS-compatible software package to allow students to explore motion planning problems and algorithms. OMPL is used in research and education; in their paper, Moll et al. describe its use to support a project-based robotics course for advanced undergraduates and graduate students. While this course does not use physical robots, students are motivated by the fact that the same code can be directly applied to real robots.

Touretzky presents 10 big ideas and questions that underlie his vision of an introductory robotics course aimed at computer science students. Touretzky argues that just as computer science students typically do not start by coding sorting algorithms in assembly, students in computer science-oriented robotics courses should not start by recreating a basic obstacle avoiding robot. The paper goes on to explore an implementation of this vision for high-level robotics and computer science through a course taught with either the Tekkotsu or ROS frameworks for programming robot behaviors. At present, these frameworks require fairly expensive robot platforms. It is our belief, and perhaps a challenge to the community, that in the age of $35 single-board computers with $25 camera modules, we may soon see a low-cost robot with the capabilities required to properly approach all 10 of Touretzky's big ideas.

We conclude the issue with Akin, Meriçli, and Meriçli's model of a first course in the fundamentals of robotics for senior computer science students. The authors show how many key concepts in robotics can be taught using a relatively inexpensive robot platform. They present a detailed week-by-week overview of a full semester course in robotics that we suspect many faculty will find to be a valuable guide to creating their own courses.

We would like to thank the reviewers for their careful and thoughtful reviews. Our reviewers were: Monica Anderson, Doug Blank, Grant Braught, Zach Dodds, Susan Imberman, Frank Klassner, Mete Kok, Fred Martin, Bruce Maxwell, Illah Nourbakhsh, Keith O'Hara, Leigh Ann DeLyser, Dave Tourtzky, and Ellen Walker. We would also like to thank the authors of all of the papers that were submitted, both those that

appear in this issue and those that we hope to see in another venue in the future.

We end this editorial with a question and an observation. First, we challenge the educational robotics community to more deeply investigate the importance of physicality: it is clear at this point that tangible, physical robots have the potential to both complicate a course as well as enhance it. Our question is not simply, "Are physical robots better than simulated ones?", but rather, "In what situations does a physical robot offer pedagogical advantages over a simulated environment?"

Finally, picking up from a thread earlier in the editorial, the field of educational robotics is seeing rapid innovation driven by advances in processing, wireless, and sensor technologies that are rapidly increasing the capabilities of educational robot platforms. In every few months, a new programmable, educational robot is released. While many of the papers in this issue specify a particular robot platform, the concepts are generally platform independent.

Jennifer S. Kay
Computer Science Department, Rowan University, Glassboro, NJ, USA
kay@rowan.edu

Tom Lauwers
BirdBrain Technologies LLC, Pittsburgh, PA, USA
tlauwers@birdbraintechnologies.com

References

Cliburn, D. (2006, June 11–15). An introduction to the Lego Mindstorms. In *Proceedings of the Association of Small Computer Users in Education Conference (ASCUE)* (pp. 25–32). Myrtle Beach, SC.

Edwards, S. (1998). *Programming and customizing the basic stamp computer*. New York, NY: McGraw-Hill.

Fagin, B. S., & Merkle, L. (2002). Quantitative analysis of the effects of robots on introductory computer science education. *Journal on Educational Resources in Computing, 2*(4), 1–17. doi:10.1145/949257.949259

Feurzeg, W. (2006). Educational technology at BBN. *IEEE Annals of the History of Computing, 28*, 18–31.

Huetinck, L., & Munshin, S. N. (2004). *Teaching mathematics for the 21st century: Methods and activities for grades 6–12*. Upper Saddle River, NJ: Pearson Prentice Hall.

Kay, J. S. (2010). Robots in the classroom … and the dorm room. *Journal of Computing Sciences in Colleges, 25*, 128–133.

Lauwers, T. (2010). *Aligning capabilities of interactive educational tools to learner goals* (Doctoral dissertation, Technical Report CMU-RI-TR-10-09). Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA.

Martin, F. G. (2001). *Robotic explorations: A hands-on introduction to engineering*. Upper Saddle River, NJ: Prentice Hall.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books.

Summet, J., Kumar, D., O'Hara, K., Walker, D., Ni, L., Blank, D., & Balch, T. (2009). Personalizing CS1 with robots. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education (SIGCSE '09)* (pp. 433–437). New York, NY: ACM. Retrieved from http://doi.acm.org/10.1145/1508865.1509018