Educational Robotics in Brooklyn

Elizabeth Sklar[1,2], **Simon Parsons**[1,2], **M Q Azhar**[2] **and Valerie Andrewlevich**[1]
[1]Dept of Computer and Information Science
Brooklyn College, City University of New York
2900 Bedford Ave, Brooklyn, NY, USA 11210
[2]Dept of Computer Science
Graduate Center, City University of New York
365 5th Avenue, New York, NY, USA 10016
sklar,parsons@sci.brooklyn.cuny.edu
mazhar@gc.cuny.edu, *valerie.andrewlevich@gmail.com*

Abstract

We describe a number of efforts to engage university students with robotics through *teaching* and *outreach*. Teaching runs the gamut from undergraduate introductory computer science to graduate-level artificial intelligence courses. Outreach involves collaborations between students and New York City public school classrooms. Our efforts have always involved team-based projects that culminate in demonstrations or competitions, usually based on challenges from *RoboCupJunior*. Several research projects have followed from these initiatives. Here, we relate some lessons learned and outline new research avenues that we are pursuing to overcome some of the issues.

Introduction

For the past five years, we have been bringing LEGO robots into university classrooms to enhance courses on introductory programming and computer science (both for computer science majors and non-majors), object-oriented programming, artificial intelligence, embodied agents and multiagent systems. We have also experimented with the use of Sony AIBO robots and are currently investigating other platforms for teaching. These experiences have led to efforts involving robotics for enriching public school classrooms through our outreach program, called **robotics.edu**.

Our initiatives have always involved team-based projects that culminate in demonstrations or competitions, frequently structured around challenges from RoboCupJunior[1]. RoboCup[2], initiated in 1997, was designed to bring together robotics and artificial intelligence researchers world-wide by providing a common problem for which a solution would require both advances in many fields and a collective approach to research within those fields (Kitano *et al.* 1997). Initially, the arena was robotic soccer, played by autonomous robots in several "leagues", distinguished by differences in physical size, hardware platform and approaches to vision and software control. This was later expanded to include robotic urban search and rescue as well. In 2000, the RoboCupJunior (RCJ) division was formed, with the goal of introducing young students (primary through high school) to RoboCup and providing them with an exciting and motivating way to learn about technology through hands-on experiences (Sklar, Eguchi, & Johnson 2002).

RCJ involves three challenges: **soccer**, **rescue** and **dance**. The soccer challenge, pictured in figure 1, is a 2-on-2 game played on a field with a floor that is landmarked using a greyscale gradient, so robots will know which direction to "kick" the ball. The ball is a special electronic sphere that emits infrared (IR) light. Robots can find the ball and determine their heading using only a light sensor; a bump sensor is also useful to prevent robots from getting stuck in corners of the field. The rescue challenge (Sklar 2004), shown in figure 2, involves robots exploring a modular, multi-level, doll-house-like structure in which white floors are marked with a black line. Robots must follow the line and locate "victims" (human-shaped figures made of green or reflective silver paper), placed strategically along the line. Teams are rewarded for accuracy and speed. The dance challenge, illustrated in figure 3, engages one or more robots in a lively event that encourages creativity. Robots move to music for a 2-minute performance that often involves costumes, scenery and even students dancing along.

Teaching

Our university teaching experiences with robotics began in Spring 2001 and have grown from one introductory robotics course for non-engineering computer science students to encompass a spectrum of courses ranging from exploring robotics for non-majors to introductory programming for majors and advanced artificial intelligence for graduate students. This section briefly outlines several of these.

Exploring Robotics

This course provides an introduction to robotics, for non-computer science majors, through the use of case studies and project-based activities. Students work together in small groups on a series of two-week creative projects, using robots to address meaningful and socially important issues, such as urban search and rescue or elder care. Along the way, students are introduced to the fundamentals of robotics (including aspects of mechanical design) and elementary programming within a graphical environment called Robo-

[1]http://www.robocupjunior.org

[2]http://www.robocup.org

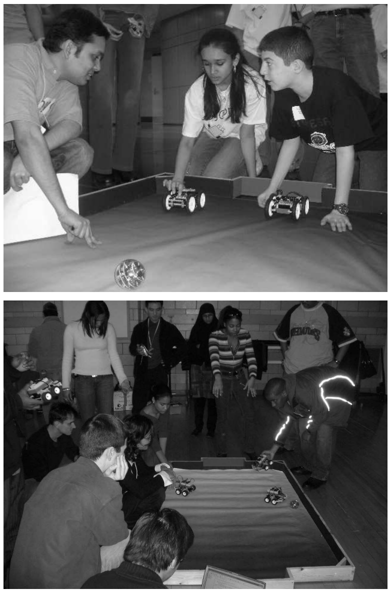


Figure 1: RoboCupJunior soccer

Figure 2: RoboCupJunior rescue

Lab[3] (Erwin, Cyr, & Rogers 2000). A series of seven scaffolded units build in complexity in terms of the robot solution, the task environment and the task(s) to be accomplished. Each unit is accompanied by a case study, with which to situate the technical material being introduced. Following is a brief outline of each unit:

1. *Introduction to Robotics:* this unit outlines basic robot construction and uses the "BigDog" project (Hambling 2006) as a case study.
2. *Simple Go-bot:* this unit introduces students to basic control ideas; the case study is the DARPA Grand Challenge (Thrun *et al.* forthcoming; Gutierrez *et al.* 2005).
3. *Dancing Go-bot:* this unit brings in touch sensors and the programming concept of *looping*, using robotic dance as a case study.
4. *Home-helper Go-bot:* this unit explains the programming concept of *branching* and the notion of event-driven programs; the case study presents the Roomba[4].
5. *Robot Teams:* this unit discusses multiple robots operating in a complex, dynamic environment and uses RoboCup Soccer as the case study; the technical challenge is RoboCupJunior soccer.
6. *Search-and-rescue Go-bot:* this unit combines touch and light sensors, making more sophisticated use of the light sensors to recognize multiple light levels. The case study is Urban Search and Rescue (USAR) robotics (Kleiner 2006), and the labs use the RoboCupJunior Rescue challenge.
7. *State-of-the-art Robotics:* this unit presents exciting new topics in the field of robotics; the case study currently being used is in the area of evolutionary robotics (Zykov *et al.* 2005).

This course has become quite popular, and currently (Fall 2006) we are offering five sections of the course with a total enrollment of 88 students.

Computing Core

This course offers an introduction to computer science and programming, for non-computer science majors, through the use of project-based educational robotics activities. The course is part of the *core curriculum* required of all undergraduates at Brooklyn College, and our department is experimentally offering several "flavors" of the course to provide a variety of interdisciplinary, applied, context-based entries into the world of computing, as part of a larger project that is attempting to broaden the demographics of students pursuing careers in computer science, particularly aiming to attract female and minority students[5]. The course is organized as above, into seven curricular units, where each unit explores a technical topic and is framed with a case study and application area for hands-on laboratory work. The curricular areas are defined by the core course, and robotics topics provide the flavor for this particular section. The areas are shown in table 1.

We are currently (Fall 2006) offering one section of this course, with an enrollment of 22 students. A formal evaluation is being conducted, with pre- and post- attitudinal surveys. In addition, a standard academic assessment for this flavor of the course will be compared with that of the other flavors and the non-flavored course; altogether, there are 29 sections of the course (three of which are flavored, and one is the robotics flavor) with a total enrollment of approximately 500 students. Formal evaluation results are forthcoming.

[3] `http://www.ceeo.tufts.edu/robolabatceeo/`

[4] `http://www.irobot.com/`

[5] `http://bridges.brooklyn.cuny.edu`

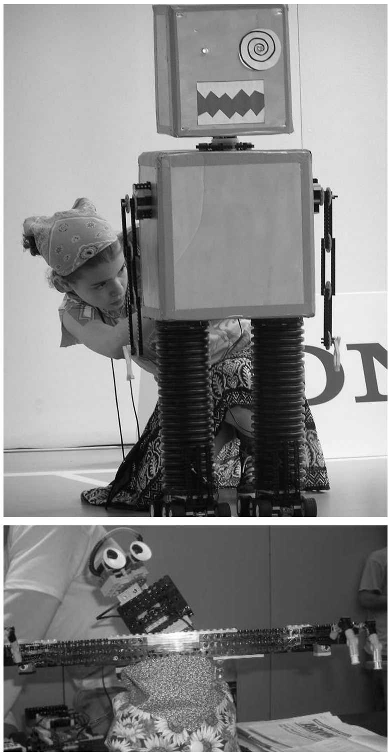


Figure 3: RoboCupJunior dance

Object-Oriented Programming

This course introduces object-oriented programming using Java and is geared toward intermediate computer science majors who have already taken two semesters of programming in C. Robotics is used as a supplemental educational tool and has been included in the course four times since Fall 2004. The course has employed the LEGO Mindstorms Robotics Invention System (RIS) and the leJOS[6] (LEGO Java Operating System), in addition to the standard (i.e., non-robotics) Java programming language. The first time we offered the course (in Fall 2004), we introduced educational robotics activities as a project towards the end of the semester, after spending the first two-thirds of the course teaching elements of the standard Java lanaguage. We received positive feedback from the students, and many said that we should introduce robotics from the beginning of the semester. During Spring 2004, we integrated robotics activities into the entire course curriculum, replacing significant portions of the take-home exercises (from the non-robotics version of the class) with hands-on lab-based robotics activities.

Every semester, we have adjusted the curriculum based on our classroom experiences, students' feedback and experiences published by others. For example, Lawhead *et al.* (2002) used Java-based LEGO robotics in an introductory programming language curriculum, and we have incorporated some of their materials which were well-suited for our class. Each semester, we have improved the logistics and academic content of the hands-on labs. We have devised pre- and post- activities for the lab sessions. The idea of the "pre-lab" is to make students' in-class lab activities more efficient. During the lab, we give students tasks to complete such as line-following, search and rescue, and RoboCupJunior soccer. The post-lab activities provide closure for the lab. In addition, we let students come to the lab outside class hours, to give them extra time with the robots if necessary. This gives students more time for problem-solving. Starting in Fall 2005, we also incorporated a "Showcase" at the end of the semester where students can demonstrate their project to their peers and invited spectators, in a RoboCupJunior soccer tournament. This event makes students more engaged in the curriculum (Beer, Chiel, & Drushel 1999).

[6] http://lejos.sourceforge.net/

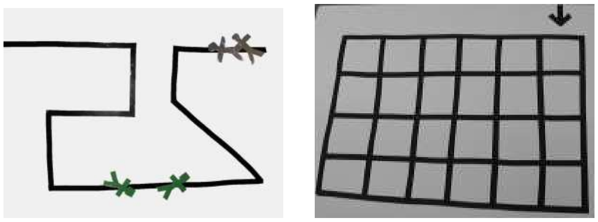
core technical topic(s)	*case study*
Introduction to Computers and Networks	tele-operated robots
Algorithms and Computer Languages	dancing robots
Machine Architecture, Data Representation and Storage	self-reproducing machines
Event-driven Programming	home-helper robots
Solvability and Feasibility	urban search and rescue robots
Programmer-defined functions	evolutionary robotics
Security, Privacy, Encryption and Plagiarism	security robots

Table 1: Topics covered in Computing Core course.

Principles of Robotics

Unlike the other courses described here, this course not only uses robots, but also is actually about robots. It is intended, for advanced computer science majors, as a broad introduction to the field of robotics, covering topics such as locomotion, kinematics, perception, localization and navigation. This theoretical background is accompanied by extensive practical work, with at least one hour of lab time for every hour spent on conventional lectures. The idea of the labs is to reinforce the main lessons explained in the theoretical work (the difficulty of navigating by dead-reckoning for example) as well as giving the students a feel for the kind of work involved in robotics research.

In previous offerings of the course, we ran the first few labs using LEGO Mindstorms RIS, as a simple platform that the students could easily master, before moving onto the more challenging Sony AIBO. In these offerings, we used the LEGO robots to perform a set of increasingly complex tasks—a race that involved line-following, some simple flocking that involved heading towards a light source, and a contest that involved line-following plus a pursuer-evader segment—before moving to the AIBO for work on naviga-



(a) RoboCupJunior rescue (b) Gridworld

Figure 4: Sample arenas

tion. Our current offering, since it includes students who have already taken courses that use the RIS, jumps right into using the AIBOs, and ends with a multi-week project.

Artificial Intelligence

The metaphor of intelligent agents is a way of bringing together the many strands of work carried out under the banner of AI and presenting them to students in a convincing way; thinking of an agent exploring an environment is a natural way to introduce search techniques, and considering how agents must respond to changes in their environments clearly shows the advantage of behavioral-based reactive techniques. The topics in our AI syllabus, which is geared toward advanced computer science majors, include: agency, control architectures, search, knowledge representation, logic, and planning.

Students engage in two robotics projects during the term, using LEGO Mindstorms RIS and the Not Quite C (NQC) language. The first project is based on a simplified version of RoboCupJunior rescue (illustrated in figure 4a). In the second project, students are confronted with a gridworld delineated with black lines (see figure 4b) where some of the squares contain the same colored figures as in the first project. The challenge is to survey the grid, identifying the positions of the figures, and then re-position the robot (at the arrow) and move to the figures in a pre-specified order in the lowest possible time. The idea behind the challenge is to bring in some of the concepts related to *search* that the students have covered in the course, combining these with the reactive techniques from the first project (which are still required to move around the grid). Since the robots cannot localize, this is a hard challenge, but it is within the capabilities of the more able students.

Further details about our experiences with using robotics in an AI class, along with evaluations, can be found in (Sklar, Parsons, & Stone 2004).

Outreach

We have established an outreach program through which undergraduate students work with in-practice teachers in New York City public school classrooms in order to introduce robotics into a number of curricular activities. Typically the undergraduates are computer science majors, and they enroll in the program through an independent internship, research project or service-learning course. While many science teachers (and others) are interested in bringing robots into their classrooms, most do not have the funding to purchase the equipment, the technical expertise to program the robots, the time to learn how to program on their own, or the curricular material to integrate robotics into their classes. We lend limited LEGO Mindstorms sets to the school for one term, after which most schools are enthralled with the program and manage to find funds to purchase their own equipment, usually 8-10 robots per classroom, depending on enrollment and age group.

Our outreach program, **robotics.edu**, pairs trained undergraduate students with classroom teachers to assist in either formal or informal learning environments. A *formal* learning environment is a structured, curriculum-oriented classroom or laboratory setting, whereas an *informal* learning environment is a less structured, non-curricular setting such as an after-school program or summer workshop. Our undergraduates, typically (though not necessarily) computer science majors, lead the class through a series of lessons that introduce students to the robots and to programming using the RoboLab environment. As part of their obligation to the project, the undergraduates are each responsible for creating and implementing a new robotics lesson at the end of the series of introductory lessons. These have included lessons on gear ratios and friction. We are gradually accumulating a database of lessons and creating a web site for sharing resources[7].

Our recent efforts have included developing and delivering materials for a 12-week technology curriculum in a “gifted and talented” middle school in Brooklyn, adapting and delivering the middle school lessons for an after-school program for inner-city girls, giving students an opportunity to explore technology which they might not otherwise experience, and developing and delivering materials for a one-day introductory workshop followed by an intense one-week summer school for inner-city middle school students in Passaic, New Jersey (a short distance from Brooklyn College), as part of a 6-year Department of Education GEAR-UP project. Each of these efforts is described briefly below.

Formal Learning in a Middle School Classroom

In Fall 2006, we introduced robotics to the technology teacher at a local “gifted and talented” middle school. Every student in the school, which covers grades 6, 7 and 8 (ages 11-14), takes a technology class each year. There are between 19-25 students, all from the same grade, in each class. The classes meet four times a week; three meetings are 45-minute periods and one meeting is a 90-minute “double” period. We have brought several undergraduate and graduate students into this classroom to support hands-on robotics lessons for several sections of this class. Note that each undergraduate (or graduate) student comes once a week, during a double period, since 45 minutes is too short for conducting effective hands-on robotics activities.

Our curriculum lasts twelve weeks, using the LEGO Mindstorms RIS and RoboLab. We have developed 5 basic lessons covering: (1) robot construction, (2) motors and motion, (3) touch sensors, (4) light sensors and (5) programming con-

[7] `http://agents.sci.brooklyn.cuny.edu/robotics.edu`

structs (branching and looping). Each lesson takes two 90-minute class sessions for the middle-school students to complete. For the last two class sessions, each undergraduate (or graduate) student is responsible for developing and delivering a lesson of their own, covering topics of their choice, such as exploring gear ratios. This is part of the requirements for their independent study course, which also includes keeping a journal of their visits to the classroom and writing up a report at the end of the term, describing their experiences and lessons learned.

For the first semester we worked with the middle school, we loaned 10 Mindstorms kits to the technology teacher, who was computer-literate but had no experience with robotics. Our students came to every class period in which hands-on robotics was taught. By the second semester, due to scheduling issues, we were only able to provide students for some of the sections in which the teacher wanted to offer robotics; so while we provided the curriculum, the teacher was comfortable enough from the first semester's experience that he was able to teach the unsupported sections on his own. By the third semester, the teacher acquired means to purchase robot kits and is currently teaching all sections plus an after-school program in robotics without regular support from our students. As a side note, at least one of the computer science undergraduates who worked in the classroom has decided to pursue a career in technology education as a result of her experience with this program.

Informal Learning in an After-School Program

Robotics offers a unique opportunity to engage girls in technology. At a primary school in Brooklyn, New York, we ran an after-school robotics program for girls in grade 6 (age 11-12) for 1 1/2 hours once a week over an 8-week period. As a way of piquing interest and creating ownership of the program, the girls were first asked to describe what meaning the word *robotics* held for each of them. Next they were instructed to create a robot of their own design with LEGO parts. Not surprisingly, they constructed something which took on a human shape and was described to accomplish tasks such as doing homework and cleaning. This was followed up with explicit instructions on building our *GoBot*. After building the four-wheeled structure, the girls developed a strong interest in the engineering aspects of the GoBot and requested the opportunity to build a three-wheeled structure. They then developed trials to see how far and fast each robot would travel under varying conditions. This was all prior to any programming being done on the robots. Finally the participants were given an introduction to programming and instruction on how to use RoboLab. This resulted in continual self-inquiry units, guided by the participants' interests and not following any set curriculum. The girls also visited RoboCupJunior while it was taking place in New York City[8] as a means of exposure to the broader participation in the world of robotics. Their enthusiasm and high level of interest in the discipline was evidenced by their unanimous expressed desire to have the after-school program continue through the end of the school year.

[8] `http://agents.sci.brooklyn.cuny.edu/rcjnyc`

Informal Learning in a Summer Workshop

Starting in Spring 2006, we have been collaborating with an inner-city school district near Newark, New Jersey (about 15 miles from New York City) as part of a project aimed at giving middle school students academic experiences that will encourage them to attend college. The 6-year "GEAR-UP" project, funded by the U.S. Department of Education, will follow a single cohort of students from middle school until they graduate from high school. We are providing educational robotics curriculum and activities for these students in a series of weekend and summer sessions that comprise the project's "Robotics Academies".

The activities introduce the students to a broad range of computer programming and robotics principles. Students collaborate in teams to design and build robots using the LEGO Mindstorms RIS and the RoboLab interface. They are given progressive, age-appropriate challenges focused on specific curricular goals and based on a discovery-based pedagogical methodology. Students learn not only about programming and robotics but also about the iterative process of designing, implementing, debugging, testing and revising that accompanies practically every engineering task. The materials we developed for the "Robotics Academies" were initially based on those we created for the "gifted and talented" middle school classroom, but we expanded them to fit the extended, intensive time periods allocated to the "Academies".

First, we held a one-day professional development course to train four classroom teachers on the LEGO RIS, the RoboLab programming environment and our curriculum, so that the teachers would be prepared to assist in sessions involving students. Then, we ran a series of three one-day "Saturday Academies" in which groups of 20-40 7th grade (age 12-13) students experienced robotics for the first time through short, hands-on activities. Students were recruited and pre-selected by teachers for each of the three "Saturday Academies". The teachers managed attendance, discipline, and other logistical details.

Finally, we ran a week-long "Summer Academy" to provide more in-depth experiences to students who were selected from the Saturday Academies. This Summer event culminated in a "Showcase" event to which parents were invited and students demonstrated their robots. The program is ongoing, and students are expected to participate in the New York City regional RoboCupJunior event in early 2007.

New Directions

Several research efforts have been inspired through our teaching and outreach activities. A multi-year evaluation project has been examining students' experiences with robotics, trying to identify what exactly students are learning when they engage with robots in a variety of settings (e.g., (Goldman, Eguchi, & Sklar 2004)). A four-year study of RoboCupJunior participants has shown that students of all ages and nationalities improve their teamwork and communication skills, in addition to their knowledge of programming and engineering (Sklar, Eguchi, & Johnson 2002; Sklar & Eguchi 2004b). Other work has examined the effect that mentoring has on the undergraduates taking part in the outreach program (Sklar & Eguchi 2004a). Some work has examined and compared teaching AI with robotics in a number of settings, including evaluations (Sklar, Parsons, & Stone 2004), showing that robotics is only helpful if the instructor ties the robotics projects directly and explicitly to the curriculum. An expanded list of the courses described here are detailed, and evaluation results presented, in (Sklar, Parsons, & Azhar 2006), which confirms the previous findings, that robotics helps motivate students, contributes to improvements in personal time management, teamwork and communication skills, but also involves increased preparation time for instructors in order to carefully and judiciously link robotics projects to curriculum so that learning stays on-track and academic-content is not sacrificed.

Some of our recent research involves development of an icon-based, graphical, universal interface and simulator for educational robotics. While the use of low-cost robotics platforms in the classroom has many attractive features, there are still several shortcomings that must be overcome in order to realize the full potential of educational robotics as a practical learning environment. Particularly since time for "practice" is limited, there is a need to reduce debugging time when using robots in instructional settings. Most robotics programming interfaces are designed for university-level or late high school students and are implemented as extensions to existing languages, e.g., (Touretzky & Tira-Thompson 2005; Blank *et al.* 2004), however such environments are not appropriate for younger students or for students for whom learning a text-based programming language is not of interest or not appropriate. We have been developing an agent-oriented, behavior-based framework designed to address some of these issues (Azhar, Goldman, & Sklar 2006). Our framework has the capability to interact with multiple agent platforms and a Flash simulator through an XML-based agent behavior language. Our longterm goal is to create a standard middle ground that can act as a "magic black box", providing a seamless transition between an icon-based graphical user interface for specifying program instructions, a simulator-based debugging environment and a range of robotic platforms, such as Sony AIBO, iRobot Roomba and LEGO NXT.

Acknowledgements

This work was partially supported by National Science Foundation (NSF) ITR #02-19347, NSF IIS #03-29037, NSF BPC #05-40549 and Department of Education (DOE) GEAR-UP #P334A050232. Thanks to Jerry Jaffee for agreeing to incorporate robotics into his middle school technology class.

References

Azhar, M. Q.; Goldman, R.; and Sklar, E. 2006. An agent-oriented behavior-based interface framework for educationa robotics. In *Agent-Based Systems for Human Learning (ABSHL) Workshop at Autonomous Agents and MultiAgent Systems (AAMAS-2006)*.

Beer, R. D.; Chiel, H. J.; and Drushel, R. F. 1999. Using autonomous robotics to teach science and engineering. *Communications of the ACM* 42(6).

Blank, D. S.; Kumar, D.; Meeden, L.; and Yanco, H. 2004. Pyro: A python-based versatile programming environment for teaching robotics. *Journal on Educational Resources in Computing (JERIC), Special Issue on robotics in undergraduate education, part I* 4(3).

Erwin, B.; Cyr, M.; and Rogers, C. B. 2000. Lego engineer and robolab: Teaching engineering with labview from kindergarten to graduate school. *International Journal of Engineering Education* 16(3).

Goldman, R.; Eguchi, A.; and Sklar, E. 2004. Using Educational Robotics to Engage Inner-City Students with Technology. In *Proceedings of the Sixth International Conference of the Learning Sciences (ICLS)*, 214–221.

Gutierrez, A.; Galatai, T.; Gonzalez, J. P.; Urmson, C.; and Whittaker, W. L. 2005. Preplanning for high performance autonomous traverse of desert terrain exploiting a priori knowledge to optimize speeds and detail paths. Technical Report CMU-RI-TR-05-54, Robotics Institute, Carnegie Mellon University.

Hambling, D. 2006. Robotic 'pack mule' displays stunning reflexes. *NewScientist.com news service*.

Kitano, H.; Asada, M.; Kuniyoshi, Y.; Noda, I.; and Osawa, E. 1997. RoboCup: The Robot World Cup Initiative. In *Proceedings of the First International Conference on Autonomous Agents (Agents-97)*.

Kleiner, K. 2006. Better robots could help save disaster victims. *NewScientist.com*.

Lawhead, P. B.; Duncan, M. E.; Bland, C. G.; Goldweber, M.; Schep, M.; Barnes, D. J.; and Hollingsworth, R. G. 2002. A road map for teaching introductory programming using lego mindstorms robots. In *ITiCSE-WGR '02: Working group reports from ITiCSE on Innovation and technology in computer science education*, 191–201.

Sklar, E., and Eguchi, A. 2004a. Learning while Teaching Robotics. In *AAAI Spring Symposium 2004 on Accessible Hands-on Artificial Intelligence and Robotics Education*.

Sklar, E., and Eguchi, A. 2004b. RoboCupJunior – Four Years Later. In *Proceedings of the Eighth International RoboCup Symposium*.

Sklar, E.; Eguchi, A.; and Johnson, J. 2002. RoboCupJunior: learning with educational robotics. In *Proceedings of RoboCup-2002: Robot Soccer World Cup VI*, 238–253.

Sklar, E.; Parsons, S.; and Azhar, M. Q. 2006. Robotics across the curriculum. Technical report, Department of Computer and Information Science, Brooklyn College, City University of New York. *(submitted for publication)*.

Sklar, E.; Parsons, S.; and Stone, P. 2004. Using RoboCup in university-level computer science education. *Journal on Educational Resources in Computing (JERIC), Special Issue on robotics in undergraduate education, part I* 4(2).

Sklar, E. 2004. A long-term approach to improving human-robot interaction: RoboCupJunior Rescue. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

Thrun, S.; Montemerlo, M.; Dahlkamp, H.; Stavens, D.; Aron, A.; Diebel, J.; Fong, P.; Gale, J.; Halpenny, M.; Hoffman, G.; Lau, K.; Oakley, C.; Palatucci, M.; Pratt, V.; Stang, P.; Strohband, S.; Dupont, C.; Jendrossek, L.-E.; Koelen, C.; Markey, C.; Rummel, C.; van Niekerk, J.; Jensen, E.; Alessandrini, P.; Bradski, G.; Davies, B.; Ettinger, S.; Kaehler, A.; Nefian, A.; and Mahoney, P. forthcoming. Stanley, the robot that won the DARPA grand challenge. *Journal of Field Robotics*.

Touretzky, D. S., and Tira-Thompson, E. J. 2005. Tekkotsu: A framework for AIBO cognitive robotics. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05)*. Menlo Park, CA: AAAI Press.

Zykov, V.; Mytilinaios, E.; Adams, B.; and Lipson, H. 2005. Self-reproducing machines. *Nature* 435(7038).