RoboCup in Higher Education: A Preliminary Report

Elizabeth Sklar[1], Simon Parsons[2], and Peter Stone[3]

[1] Department of Computer Science
Columbia University
1214 Amsterdam Avenue, New York, NY 10027, USA
sklar@cs.columbia.edu

[2] Department of Computer and Information Science
Brooklyn College, City University of New York
2900 Bedford Avenue, Brooklyn, NY 11210, USA
parsons@sci.brooklyn.cuny.edu

[3] Department of Computer Sciences
The University of Texas at Austin
1 University Station C0500, Austin, TX 78712-1188, USA
pstone@cs.utexas.edu

Abstract. Since team-based projects have been proven to be an effective pedagogical tool, we have been using RoboCup challenges as the basis for class projects in undergraduate courses. This paper unifies several independent efforts in this direction and presents early work in the development of shared resources and evaluation. We outline three courses and describe the related class projects in order to make the context of our investigation clear and to make it possible for others to replicate or extend our work, and contribute to the shared resource.

1 Introduction

Educational robotics, the use of robotics as a form of hands-on learning environment, is becoming increasingly common as robot kits are becoming more accessible and affordable [31]. Creative instructors are finding ways to teach science topics using these technologies, organizing tournaments around the robots; the energy, enthusiasm and motivation displayed by students of all ages is unsurpassed. We have found RoboCup — especially Soccer Simulation and the RoboCupJunior challenges — to be particularly conducive to college-level classroom use. The ability to demonstrate theoretical models and complex algorithms with a hands-on, accessible medium strengthens the learning experience.

In this paper, we document our experiences incorporating RoboCup activities into undergraduate courses with the idea of uniting others who are doing the same. Our aim is two-fold: one, to create a repository for related curricular materials; and two, to build a common instrument and database for evaluating the RoboCup learning environment. While we have found the link between RoboCup and traditional coursework in Introductory Robotics, Artificial Intelligence and Multiagent Systems to be a natural one, we presume that this arises out of our familiarity with RoboCup through longterm involvement with the initiative. In developing our repository, we are hoping to make the notion of incorporating RoboCup into such coursework a relatively easy task for uninitiated instructors, by providing syllabi, reading lists and project descriptions.

Others have experimented with the RoboCup paradigm in undergraduate classrooms. Coradeschi and Malec used the RoboCup soccer simulator in a course on Artificial Intelligence Programming[1] [11]. Birk has developed a course on Autonomous Systems that uses the Small-Size RoboCup League for practical exercises[2]. Vidal and Buhler [34] have developed a series of graduate level courses on multiagent systems using the RoboCup Simulation league.

We are also, of course, not the first to use robot kits in an undergraduate classroom as a hands-on learning environment. In 1989, Martin created the MIT Robot Design project course (6.270) [16, 18]. Yanco [36] has adopted this course, ending the term with a Botball[3] tournament. Mataric's "Introduction to Robotics" [20] takes a hands-on approach to the introduction of the basic concepts in the field of robotics and concludes with a contest where robots play a ball game in a hexagonal field. There are also courses using hands-on robotics that do not focus on teaching robotics as the main subject. Littman's "Programming Under Uncertainty" [15] teaches about methods for programming under uncertainty and a variety of machine learning techniques.

Aside from constructing a shared repository of course materials, we are interested in conducting a comprehensive evaluation of the pedagogical value of educational robotics in general and RoboCup activities in particular. Following on the work of Sklar et al. [30], we are interested in trying to pinpoint the educational value of robotics and the RoboCup initiative at the undergraduate level. Uniting multiple instructors means that we can not only share experiences, but also collect course evaluation data on a grander scale, allowing us to perform analysis across a broader cohort, with a range of academic as well as cultural backgrounds.

2 Robotics in undergraduate education.

Here, we describe three classes where we have successfully used RoboCup challenges as term projects.

2.1 Introduction to Robotics.

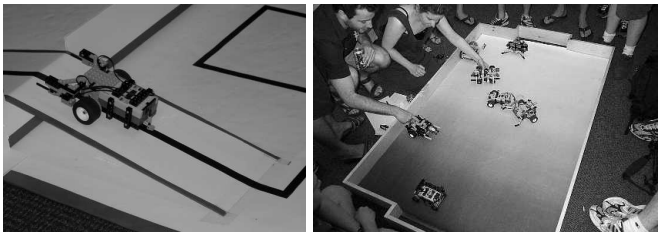
This introductory course looks at robotics from several aspects: technically, historically and socially. Many of the technical aspects are based on Mataric's course described above. The course is designed for non-engineering students to gain a hands-on experience with technology, as well as a basic understanding of the field of robotics and the challenges facing the field today. Part of the course is spent reading and discussing classic material that relates to robots — including non-technical aspects such as science fiction, psychology, cognitive science and education. The remainder of the course takes a hands-on approach to introducing the basic concepts in robotics, focusing on autonomous mobile robots. LEGO Mindstorms robots[4] are used, and students must complete two projects with them. First, they must build robots to execute a line-following task culminating in a maze contest. Second, they construct robots to play soccer and perform in a RoboCupJunior style two-on-two tournament.

[1] http://www.ida.liu.se/~silco/AIP/

[2] http://www.faculty.iu-bremen.de/birk/lectures/COURSES/autosys.html

[3] http://www.botball.org

[4] http://www.legomindstorms.com



(a) Line-following maze (b) Soccer

Fig. 1. Robot contests.

The course begins with an introduction to robotics and agent-based artificial intelligence [28]. Then we cover some history and the basics of building and programming with LEGO Mindstorms [17, 19], using Not Quite C[5] [4, 5]. The robots are used as examples for the remainder of the topics, which introduce the general areas in robotics: effectors, sensors and control [18, 23]. The area of control is covered in more depth, discussing various architectures including deliberative, reactive, hybrid and behavior-based [1, 6, 8, 22]. Learning is also discussed [13, 21, 35]. Other topics presented include artificial life [3, 10], edutainment [14, 32], cognitive science and psychology [7, 24] and science fiction [2].

The course is taught over a 14-week semester. There is one 75-minute lecture and one 75-minute lab per week. There are two exams, and students submit written lab reports documenting their software and hardware developments. They are encouraged to record results of tests made and changes to their designs. Students also prepare a term project, presented both written and orally.

Sklar taught this course in Spring 2001 at Boston College[6]. Twenty-seven students were enrolled, three of whom were female. All were undergraduates, and there was a mix of ages: first year (1 student), second year (3), third year (10) and fourth year (13). The Computer Science Department at Boston College is in the School of Management and there is no engineering school in the university, so the hands-on technical experience of these students was limited. Sixteen members of the class were Computer Science or Information Technology majors. The rest came from Biochemistry (1 student), Communication (1), Economics (4), History (2), Marketing (1), Mathematics (1) and Physics (1).

The students were placed in groups of three for working on the robotics projects. Since the experience levels of the class was so diverse, Sklar assigned the groups, attempting to balance each group with an equal number of beginning and advanced students. Students were given some lab time during the scheduled course period in order to work on the projects. However, this was not enough time to perfect robots to perform well in the contests, so many of the students met outside of class time to work on the robots. Each student was required to submit a lab report individually, which included an assessment of the contribution of their teammates. The efforts of team members are never balanced, however the inequities were obvious in reviewing the lab reports, even

[5] `http://www.baumfamily.org/nqc/index.html`

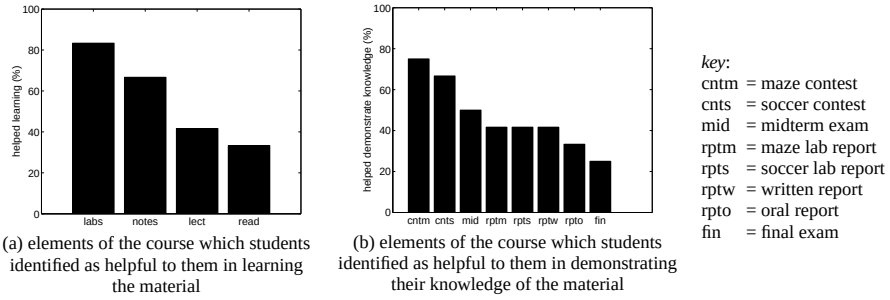
[6] `http://www.cs.columbia.edu/~sklar/teaching/spring2001/mc375/default.html`

without the peer assessment component. Students' grades were based on the lab reports, not on their robots' performance in the contests.

The two contests were held in a public space and students were encouraged to invite their friends to watch. The excitement of the crowd and the visibility of the event motivated students to work harder after the first (maze) contest in preparing for the second (soccer) contest.

The term projects presented a major challenge for these students, who were not typically asked to do any writing in Computer Science classes. They were required to submit a brief project proposal several weeks prior to the final due date, in order to get them started and also to provide feedback about the appropriate nature of the topic. The range of topics chosen was quite broad, from the use of nanotechnology in surgical robots to the history of robots dating back to ancient Greece. Each student gave a ten-minute oral presentation on their chosen topic. This was difficult for many students who were not used to speaking in front of a class. Although discussion following the presentations was encouraged, very little actually occurred and several students skipped class on presentation days when they were not speaking. Course evaluation results (below) confirmed that the motivation surrounding the term project was minimal.

Students were given a survey at the end of the course. Forty-four percent of the class responded. The survey collected demographic data and also queried the students about their learning experience. They were asked to identify which elements of the course were helpful in learning the material and which elements of the assessment were valuable in helping them to solidify and demonstrate their knowledge of the subject. The results are shown in Figure 2.



(a) elements of the course which students identified as helpful to them in learning the material

(b) elements of the course which students identified as helpful to them in demonstrating their knowledge of the material

Fig. 2. Survey Results from Introduction to Robotics course, Spring 2001.

Overwhelmingly (83%), the students felt that the labs (i.e., building and programming the robots) was helpful for learning the material, whereas only 33% said that the reading was helpful. Seventy-five and sixty-seven percent responded that the two contests (maze and soccer, respectively) were valuable in helping them solidify and demonstrate their knowledge of the material. This confirms our intuition that the hands-on components provide more effective learning experiences than other aspects of coursework, particularly at the introductory level. We speculate that the readings chosen were perhaps too advanced for most of the class.

Student comments were overall quite positive, including the following statements:

– “Great course... loved the lax atmosphere and hands-on experience. I’d recommend the course to any CS major.”
– “I think the class idea is great. It is a great hands-on experience to try out. The labs were very fun times.”

There were many comments that the mixed age group was helpful for all students, as the inexperienced students learned from the more advanced, and in assisting others, the advanced also learned more themselves. Negative remarks centered around requests for more lab time and less time spent on oral presentations.

2.2 Artificial Intelligence.

The modern view of Artificial Intelligence (AI) [28] is that it is the study of intelligent agents — autonomous computing systems that perceive their environments and act upon them in a way that both responds to changes in their environments and works towards underlying goals. Robots are prototypical agents that have to move around, and react to, their environments in pursuit of their goals. Thus it is highly appropriate to explore areas of a typical artificial intelligence syllabus using robotics projects.

This course is designed to give a broad understanding of the basic techniques in use today for building intelligent computer systems. The syllabus broadly follows the outline of Nilsson’s *Artificial Intelligence: A new synthesis* [25]. Students learn about state-space representations, problem reduction, means-end analysis, and reinforcement learning. They study search methods including depth-first, breadth-first and best-first search, as well as hill-climbing and alpha-beta pruning. Predicate calculus is introduced, along with various methods of theorem proving. The course is taught over a 14-week semester, with two 75-minute lectures a week, two exams and two robotics projects.

Parsons taught the course for the first time at Columbia University in Spring 2002[7]. Thirty-five students were enrolled, of whom 6 were female. There was a wide range of students taking the course — the bulk were undergraduates (68% of the 19 for whom we have this data), but there were also 6 graduate students, both Master’s students and PhD students, and even within the undergraduate students there was a mix of ages from first year (1 student), second year (1), third year (8) and fourth year (4). The majority of the undergraduates were Computer Science majors (74%). The rest were from biology (1 student), economics (1), electrical engineering (1) and mechanical engineering (2).

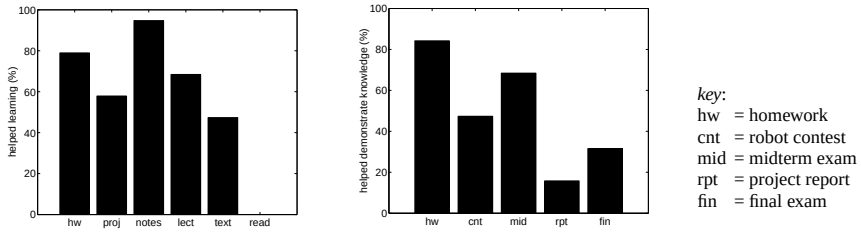
The students formed themselves into groups of three to four and had to program LEGO robots in the Not Quite C programming language to perform two tasks — a RoboCupJunior style line-following rescue task (in which the robot had to follow a convoluted line, detect an obstacle and back-up, climb and descend a gradient, and finally detect and head towards a light source) and to play a simplified version of the RoboCupJunior soccer task (the robot started at one end of a standard RoboCupJunior two-on-two soccer pitch, with the ball at the halfway point, and the robot had to manoeuvre the ball into the opposite goal, rather like a penalty kick into an empty goal). The culmination of the project was a contest in which the basis for competition was the cumulative time taken to complete both tasks, and the students also wrote a report on

[7] `http://www.cs.columbia.edu/~sp/4701-2.html`

the project. Students were also given the option of an extra-credit project of building a dancing robot exactly as in the RoboCupJunior dance competition.

Two course evaluations were administered. One was an official evaluation done by the engineering school. The other was an informal paper-and-pencil survey given out in class. The results of the engineering school's evaluation showed that 55% of the 33 students who responded gave the robotics project they undertook a rating of 5 (on a 5-point scale) for interest, and two-thirds gave it a rating of 4 or 5. Twenty-one percent of the same cohort of students gave the project a rating of 5 for the amount learned during the project, and 58% rated it 4 or 5.

The informal survey probed more into the students' perception of the value of the project as opposed to other aspects of the course. In particular, students were asked to identify which aspects of the course most contributed to helping them learn the material, and which aspects were most helpful to them in demonstrating knowledge of the material. The results are given in Figure 3. These show that the students felt that the project work was not as helpful in learning as some of the more traditional aspects of Computer Science courses, but was more useful than the textbook and additional reading material (which no students felt were useful). The picture is much the same for the demonstration of knowledge, with students rating the contest as more helpful than the final, but less helpful than the homework and midterm. The report was rated least useful of all (they really hated having to write a report). Despite the rather unencouraging figures from this second survey, the very obvious enjoyment that the majority of the students took in the projects encouraged us to repeat the experiment the following semester.



(a) elements of the course which students identified as helpful to them in learning the material

(b) elements of the course which students identified as helpful to them in demonstrating their knowledge of the material

Fig. 3. Survey Results from Artificial Intelligence, Spring 2002.

The second offering of the course was given by Parsons at Brooklyn College, City University of New York (CUNY) in Fall 2002[8]. With the exception of the absence of Masters and PhD students, the cohort was broadly similar to that at Columbia in terms of the factors we measured. Eighteen students were enrolled, of whom 7 were female. Again there was a wide range of students taking the course — all were undergraduate students but there was a considerable mix of ages with 2 second year students, 3 third year, 10 fourth year and 2 fifth year students[9]. The majority of the undergraduates were

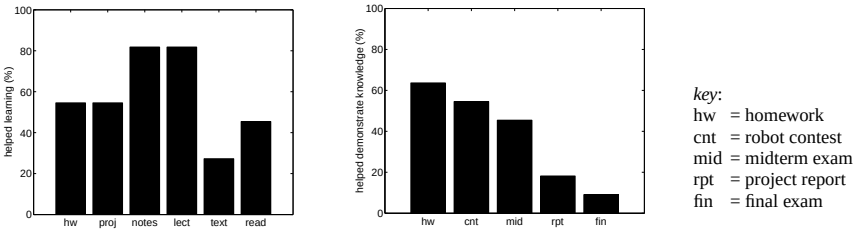
[8] `http://www.sci.brooklyn.cuny.edu/~parsons/courses/cis32-fall-2002/`

[9] It is common for CUNY students to take more than four years to complete since many study part-time.

Computer Science majors (16 students). The remaining students were from Political Science and Business.

In this offering, the project was given much later (due to problems with access to the robots) and while the teams were the same size (3 students), the project took the form of just the line-following exercise described above. Again the project ended with a contest and there was an extra-credit option to build a dancing robot.

Once again, an unofficial survey was administered, and the results are presented in Figure 4. These results are a little more encouraging than those from Columbia. This time the project was still felt to be less helpful in learning than lectures or lecture notes, but on a par with homework and more helpful than additional readings or the textbook. In terms of demonstrating knowledge, the students felt that the project was more helpful than either midterm or final. While gratifying, these figures should be viewed with some suspicion. First of all, these students had self-selected to do robot projects (students were allowed to do a non-robotics project instead if they preferred). Second, as a result of the other project, these figures are based on a very small sample of just 11 students. Finally, it seems that some of the reaction is because it is so unusual for students at Brooklyn College to get to do project work (in the open comment part of the survey several confessed that this was the only project they had ever done). The effect of this influence is supported by the fact that broadly similar results were generated by students who did the non-robotics project (though the very small number of students in this category makes the results extremely unreliable and so they are not presented here).



(a) elements of the course which students identified as helpful to them in learning the material

(b) elements of the course which students identified as helpful to them in demonstrating their knowledge of the material

Fig. 4. Survey Results from Artificial Intelligence, Fall 2002.

Comments from the student surveys from both offerings of the course include:

- "When working with the robot, I learnt that nothing is perfect in the real world. A lot of times the outcome is very unexpected."
- "It reminded me of why I want to stay away from hardware as much as possible."
- "It helped immensely! It helped me understand some of the concepts covered in the lecture."

2.3 Autonomous Multiagent Systems

Autonomous agents and multiagent systems (AAMAS) is one of the fields to which RoboCup participants have contributed consistently and prominently over the years. Despite being the basis for a large subfield of AI, there is no generally accepted definition of artificial intelligence *agents*. In loose terms, agents are programs that (i) sense

their environment, (ii) make decisions about how to act based on these sensations, and (iii) then execute these actions. Autonomous agents do all three of these steps on their own, i.e., without a human in the loop. Multiagent systems are collections of multiple agents that interact with one another. The field of AAMAS covers a wide variety of research foci and applications, including software-based information processing, robotic control of multiple agents, entertainment agents and tutoring agents [12].

This course provides a broad introduction to autonomous agents with an emphasis on multiagent systems. Topics include agent architectures, inter-agent communication, agent teamwork, distributed rational decision making, agent modeling, multiagent learning and entertainment agents. In addition to teaching about AAMAS, the course aims to introduce undergraduates to the full spectrum of research activities engaged in by professional computer science researchers, emphasizing the difference between these activities and the activities of a typical undergraduate student [33]. As such, the course includes an open-ended programming project, readings from the research literature, public speaking and writing requirements, and opportunities to collaborate with peers. In order for students to succeed, they need to attain a mastery of the AAMAS subject. However assessment is based primarily on their ability to engage in the full range of activities required of researchers.

The central focus of the course is a semester-long build-up towards a class robotic soccer competition in the RoboCup Soccer Server [26]. Students are assigned a series of four preliminary programming assignments designed to get them familiar with Soccer Server and the CMUnited client code [27]. By the end of these preliminary assignments, they have created a fully functional team (although not one that is particularly competent). The students are then encouraged to propose an improvement on this team as the topic for their final projects[10]. For example, one student proposed to use machine learning techniques to train a good goaltender without paying attention to the rest of the team.

The majority of the readings for this course are primary sources chosen both to introduce particular topics and to engender some controversy (e.g., reactive [9]) versus deliberative [29] agent architectures). To encourage the students to complete the readings in a timely fashion, they are required to submit a brief written answer to a single question pertaining to the readings at least 2 hours before the class starts. The fact that the responses are due two hours before class allows the instructor to incorporate them into the class discussion. Another effect of the questions is that the students *do* come to class prepared to discuss the readings. As a result there have been many extended and heated class discussions.

An important component of class participation is that each student is required to moderate at least one class discussion pertaining to that week's readings. They are instructed to either defend a controversial statement or pose a question and be prepared to defend either side depending on how the class reacts. This activity turns out to be one of the most difficult for the students to complete. Many of them are not used to speaking in front of a class, and they have rarely been put in the position of *facilitating* discussions as opposed to defending specific positions.

[10] They are also given the option to propose a programming project in a multiagent domain of their choice, but typically few students choose to do so.

The course requires a good deal of writing from the students. As above, the students are required to provide weekly written responses to questions related to the readings. They receive feedback pertaining to the clarity and soundness of their responses. More significantly, the students are required to write three documents pertaining to their final projects. First, they write project proposals defining their goals for their projects as well as the proposed methods for achieving them. Second, they revise their proposals and add a section on their work in progress to create progress reports. Finally, they write final reports in the format of conference papers.

Students optionally work in pairs on the final project. Teams must write their proposals and reports individually, with clear indications of what role each person played in the collaboration; and as such, more is expected from them as a final product. The robotic soccer project lends itself to such collaboration nicely since there are many different ways in which the students can divide up the work.

The class culminates in a simulated soccer tournament. The students are told at the outset that performance in the tournament will have no negative impact on their grades (while a strong performance *can* have a positive impact). Nonetheless, the tournament is a strong motivational factor for the students. Visitors are invited to the event and the students present their approaches orally and field questions as their teams are playing. The performance spread among the teams is often very large, especially given the fact that some students do not focus on creating winning teams. All class champions have been tested against a mid-range RoboCup entry and lost significantly: despite starting with a fairly detailed client code base, the students are not able to attain competitive world-class levels. However, given their time limitations this fact is neither surprising nor discouraging.

Stone has taught this course twice, first at New York University in the Fall of 2001. Fifteen students were enrolled, only one of whom was female. All students were Computer Science majors. All were graduate students: 12 masters and 3 Ph.D. The second offering of the course was at the University of Texas at Austin during Fall of 2002[11]. Again, fifteen students were enrolled, however none were female. This time, the cohort were undergraduates. Fourteen were seniors (fourth year) and one was a junior (third year). Most students were Computer Science majors, with the remainder majoring in Computer Engineering.

Course evaluations and surveys were administered at the conclusion of both courses. At NYU, the course was rated 4.5 out of a possible 5 (highest rating). At UT Austin, the course was rated 4.6 out of a possible 5. Student comments have also provided evidence that the students appreciate the opportunity to be exposed to the various components of scientific research. Both times the course has been run, at least one student has described the course in graduate school applications as a primary motivation for going on to do research. In addition, two students from the Fall 2002 offering are actively contributing to the UT Austin entry in the RoboCup 2003 competition. In one case, the research is leading up to the student's senior thesis.

An informal survey was administered in the middle of the term to the UT Austin cohort. Students were asked to rate the programming assignment on a scale of 1 to 5. Thirty-three percent gave it the highest rating; 57% gave it the second highest, while

[11] `http://www.cs.utexas.edu/~pstone/Courses/378fall02`

10% scored it average and no students entered low marks. Students also rated the reading assignments on the same scale. Twenty percent gave the highest rating; 60% gave the second highest, 13% scored it average and 7% gave the lowest rating.

Some of the comments from students have included:

- "The discussions we have in class are quite unique, I haven't had such involving discussions in any class before;"
- "Format of the class is perfect. I've waited through three years of college for a class like this.. . . I like the class so much that my other classes now disappoint me;" and
- "The only thing I dislike about the class is that we are limited in our application of our knowledge. Our education in AI is directed at implementing a RoboCup soccer agent. I feel that if we were able to apply our knowledge to other aspects, we would gain an even better understanding of artificial intelligence."
- "The simulator code was kinda tricky to understand."

3 Discussion.

The three courses described above offer an interesting comparison, not only in terms of content and presentation but also in regard to the cohorts of students enrolled. Collectively, the courses have been offered five times at five different universities, providing a broad range of backgrounds, demographics and experience levels — from first-year undergraduates at a private, non-engineering college to fourth-year undergraduates at a large state university, and including graduate students from both a public city university and two large private universities. Thus the positive feedback across the board in regard to the robotics projects is an encouraging and significant factor.

Comments about the reading materials were typically less enthusiastic, as the figures presented in the previous section indicate. Stone's course, where students were required to respond to the reading in short written assignments and were then given the opportunity to discuss the readings formally in class, fared better than the other courses, where reading was assigned in a more traditional manner — without written reading responses and primarily the material was presented by the instructors in lectures where questions were encouraged (but infrequent) and discussion was not the central theme. Comparatively, Stone's classes had fewer students, so more effective discussion was possible. Nonetheless, several students from all the courses commented that they wished there had been better connections between the readings and the project work. This type of feedback is valuable to us and our colleagues in improving the existing courses as well as designing new ones.

The challenge presented to students who were required to make oral presentations to the classes (in both Sklar's and Stone's courses) is also notable. No matter what career path is ultimately taken, students need to know how to communicate their ideas. The development of oral presentation skills is important and should be encouraged, despite students' dislike of this aspect of the courses. Perhaps more creative ways of oral reporting can be incorporated into all the courses.

The evaluations performed on Stone's course so far is limited to generalized questions about whether students liked the course and standard questions about the instructor and workload. While this level of information is useful to administrators, we are

interested in gathering more specific data on the students' learning experiences. The discrepancy in evaluation methodology from one course offering to another is one of the factors that has spurred us to create the repository mentioned here. This will include a standard instrument for measuring the effectiveness of specific coursework and the general RoboCup learning environment.

4 Summary.

We have presented our development of a repository for educational robotics activities, particularly focused on RoboCup challenges. Our goal is to build an on-line space for sharing curricular materials and to develop a unified instrument and database for evaluating the RoboCup learning environment. As examples of the type of information we are seeking and archiving, we have presented an account of our collective experiences incorporating RoboCup activities into undergraduate courses.

We will continue our efforts with the existing courses described above as well as development of new courses. The Autonomous Multiagent Systems course will be offered in Fall 2003 by Parsons at the CUNY Graduate Center and in Spring 2004 by Stone at the University of Texas at Austin[12]. Sklar is currently adapting her course to an introductory computer science curriculum, using the robotics as a basis for demonstration. The survey instrument will be adapted to both courses and administered at the end of each term.

We hope to encourage others to join in this community venture. Our on-line repository can be found evolving at `http://agents.cs.columbia.edu/er`. We welcome contributions and participants.

References

1. R. C. Arkin. *Behavior-Based Robotics*. MIT Press, 1998.
2. I. Asimov. *I, Robot*. Doubleday, Garden City, NY, 1950.
3. T. Balch, Z. Khan, and M. Veloso. Automatically Tracking and Analyzing the Behavior of Live Insect Colonies. In *Proc of Agents-01*, Montreal, 2001.
4. D. Baum. *Dave Baum's Definitive Guide to LEGO Mindstorms*. APress, 2000.
5. D. Baum, M. Gasperi, R. Hempel, and L. Villa. *Extreme MINDSTORMS: An Advanced Guide to LEGO MINDSTORMS*. Apress, 2000.
6. A. Birk. Behavior-based robotics, its scope and its prospects. In *Proc of 24th IEEE Industrial Electronics Society Conf*. IEEE Press, 1998.
7. V. Braitenberg. *Vehicles: Experiments in Synthetic Psychology*. MIT Press, 1984.
8. R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1), 1986.
9. Rodney A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47, 1991.
10. A. Colorni, M. Dorigo, and V. Maniezzo. An Investigation of some Properties of an Ant Algorithm. In *Proc of PPSN-92*. Elsevier Publishing, 1992.
11. S. Coradeschi and J. Malec. How to make a challenging AI course enjoyable using the RoboCup soccer simulation system. In *Proc of RoboCup-98*. Springer Verlag, LNAI 1604, 1998.
12. Maria Gini, Toru Ishida, Cristiano Castelfranchi, and W. Lewis Johnson, editors. *Proc of AAMAS-02*. ACM Press, 2002.

[12] `http://www.cs.utexas.edu/~pstone/Courses/378spring04`

13. I. Harvey, P. Husbands, and D. Cliff. Issues in Evolutionary Robotics, Cognitive Science Research Paper Serial No. CSRP 219. Technical report, Univ of Sussex, Brighton, UK, 1992.
14. H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. RoboCup: The Robot World Cup Initiative. In *Proc of Agents-97*, 1997.
15. M. Littman. CPS196: Programming Under Uncertainty. `http://www.cs.duke.edu/~mlittman/courses/cps196/`.
16. F. Martin. 6.270: The MIT LEGO Robot Design Project Competition. `http://fredm.www.media.mit.edu/people/fredm/projects/6270/`.
17. F. Martin. Kids Learning Engineering Science Using LEGO and the Programmable Brick. In *Proc of AERA-96*, 1996.
18. F. Martin. *Robotic Explorations: A Hands-On Introduction to Engineering*. Prentice Hall, 2000.
19. F. Martin, B. Mikhak, M. Resnick, B. Silverman, and R. Berg. To Mindstorms and Beyond: Evolution of a Construction Kit for Magical Machines. *Robots for Kids: Exploring New Technologies for Learning*, 2000.
20. M. Mataric. CS 445 Introduction to Robotics, a LEGO-kit-based hands-on lab course. `http://www-scf.usc.edu/~csci445/`.
21. M. J. Mataric. Reward Functions for Accelerated Learning. In W. W. Cohen and H. Hirsh, editors, *Proc of Machine Learning Conf*. Morgan Kaufmann Publishers, 1994.
22. M. J. Mataric. Behavior-Based Control: Examples from Navigation, Learning, and Group Behavior. *Journal of Experimental and Theoretical Artificial Intelligence, special issue on Software Architectures for Physical Agents*, 9(2-3), 1997.
23. P. J. McKerrow. *Introduction to Robotics*. Addison-Wesley Publishing Co., 1991.
24. M. Minsky. *Society of Mind*. Picador, 1987.
25. N. J. Nilsson. *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann, 1998.
26. Itsuki Noda, Hitoshi Matsubara, Kazuo Hiraki, and Ian Frank. Soccer server: A tool for research on multiagent systems. *Applied Artificial Intelligence*, 12, 1998.
27. Itsuki Noda and Peter Stone. The RoboCup soccer server and CMUnited clients: Implemented infrastructure for MAS research. *Autonomous Agents and Multi-Agent Systems*, 7(1&2), July 2003. To appear.
28. S. J. Russell and P. Norvig. *Artificial Intelligence: a Modern Approach*. Prentice Hall, 1995.
29. Reid Simmons. Structured control for autonomous robots. *IEEE Transactions on Robotics and Automation*, 10(1), Feb 1994.
30. E. Sklar, A. Eguchi, and J. Johnson. Robocupjunior: learning with educational robotics. In *Proc of RoboCup-2002*, 2002.
31. E. Sklar and S. Parsons. Robocupjunior: a vehicle for enhancing technical literacy. In *Proc of AAAI-02 Mobile Robot Workshop*, 2002.
32. E.I. Sklar, J. Johnson, and H.H. Lund. Children Learning from Team Robotics: RoboCup Junior 2000 Educational Research Report. Technical report, The Open Univ, UK, 2000.
33. Peter Stone. RoboCup as an introduction to CS research. In *Proceedings of the RoboCup-2003 Symposium*, Padova, Italy, July 2003.
34. José M. Vidal and Paul Buhler. Teaching multiagent systems using RoboCup and Biter. *The IMEJ of Computer-Enhanced Learning*, 4(2), 2002.
35. R. A. Watson, S. G. Ficici, and J. B. Pollack. Embodied Evolution: Embodying an Evolutionary Algorithm in a Population of Robots. In *Proc of CEC-99*, 1999.
36. H. Yanco. 91.450: Robotics I. `http://www.cs.uml.edu/~holly/91.450/`.