Extra-curricular Robotics: Entry-level Soccer for Undergraduates

Susan Imberman[1] **Aleksandr Barkan**[2] **Elizabeth Sklar**[2]
[1]Dept of Computer Science, College of Staten Island, City University of New York
2800 Victory Blvd., Staten Island, NY 10312 USA
[2]Dept of Computer and Information Science, Brooklyn College, City University of New York
2900 Bedford Ave., Brooklyn, NY 11210 USA0
imberman@mail.csi.cuny.edu, aleckz@gmx.net, sklar@sci.brooklyn.cuny.edu

Abstract

In this paper we describe an extracurricular approach to experimenting with robotics. We argue that university computer clubs are a good place for students to experience robotics outside the computer curriculum. Robot competitions are one way in which computer clubs can become involved with robotics. Some venues are more suitable than others for student university computer organizations. We describe the current state of RoboCup-inspired *ELeague* soccer and why it is a good match for these types of organizations.

Introduction

The integration of robotics into the computer science curriculum in courses such as CS1 (Fagin 2004; Imberman 2005), Artificial Intelligence (Greenwald 2004; Imberman 2004; Parsons 2004; Sklar *et al*, 2004), as well as stand-alone robotics courses (Dodds 2004) has been well documented in the literature by a wide range of experience reports and some research reports. Students are, by all accounts, interested and motivated by these hands-on learning experiences.

However, there are also many drawbacks to using robots in the curriculum which have been cited in the literature. One of the primary drawbacks recounted in nearly every report is that adding robots to coursework takes time away from the prescribed syllabus. Despite attempts by instructors to provide pre-constructed robots, or instructions for basic robot constructions, students are usually distracted by the types of mechanical, engineering, hardware and general "real-world" problems that plague robotics researchers.

A secondary drawback is the constraints placed on students to do their homework in laboratory environments, and on instructors to provide access to working equipment such that there is enough time for even the slowest students to complete their assignments. While some innovative and/or well-resourced instructors are able to overcome this secondary limitation, researchers are actively working on more widely applicable solutions to help large, less-well-funded institutions, where it may be impractical to provide robots and laboratory time to accommodate all students.

A third drawback involves the costs in providing robotic equipment. Courses with large numbers of students can require a substantial amount of equipment. In addition, the high cost of most robot systems precludes many students from purchasing and exploring on their own. Having robots available for students to experiment with, outside the classroom may be unattractive to Computer Science departments for reasons of cost and wear-and-tear on existing robot equipment.

Methods that allow students to experiment with robotics outside the traditional classroom are needed to fill this gap. Some typical means for accomplishing this are through academic research projects for course credit or funded programs like research experience for undergraduates (REU). However, these are considered formal in that students are generally required to write research reports at the end of a semester and participation may be limited to one independent study course or availability of funding.

Our interest here is to explore more informal settings for robotics experiences, through extra-curricular "clubs". Most colleges have student clubs that reflect the varied interests of their students. Computer science clubs are prevalent in many of these colleges. Within the 19 campuses that belong to the City University of New York (CUNY) system, nine four-year and two-year institutions advertise some type of computer-oriented club.

It is not unusual for a student club to participate in a competitive venue. Programming contests, such as the ACM programming contest, are popular activities for many computing clubs. Although organizing a programming team is easier with respect to equipment acquisition, robotics competitions have an intrinsic appeal.

Our focus in this paper is to present the *ELeague* robot soccer (Anderson *et al.* 2003), which was modeled after the RoboCup F-180 league as an appropriate activity for informal extracurricular robotics. We describe our motivations for this project in the next section. Existing robot competitions are then examined. The section following describes how one might organize a computer science club for robotic competition, along with some of the issues and difficulties involved. The technical implementation details of the league are then discussed; and the paper closes with a discussion and conclusions.

Motivation

Our main motivation is to offer a way for students to become more involved in robotics via intercollegiate competitions, in much the same way that intramural competitions occur in athletics. Pedagogically speaking, academically oriented clubs provide a readily available group of students who might be interested in competitions of this nature. One need only look at the popularity of high school robotics competitions to see that the interest exists.

High school level competitions are popular for several reasons. There is usually a wide breadth of resources available for competition participants, making entry into the competition easy. Teachers tend to be highly motivated. Grants available to schools for equipment purchase, helps mitigate the expense. Parental involvement provides both social and financial support. In addition, college bound students see participation in a robotics competition as "looking good" on college applications.

While many of the same motivators exist at the college level, grant money for robot competitions is virtually nonexistent. Most colleges collect some kind of "student activities fee", but monies provided to individual student organizations are usually not enough to support a robotics club. Parental involvement is usually non-existent at the college level.

As well, there are differences between organizing on the college level versus the high school level. In order for informal extracurricular robotics at the college level to succeed, several things need to be in place. First the resources needed for the competition need to be usable with little start-up effort; students in academic clubs need more instant gratification since they are participating on their own time (not for academic credit or for pay). Essentially there are no extrinsic motivational aspects, therefore there needs to be enough intrinsic motivational aspects to get them in the door the first time and keep them involved. There is also need for an involved community, similar to that found in RoboCup and RoboCupJunior, to organize competitions, formalize rules, etc.

Competitions

Competitions have long been a mainstay in the robotics community. Well-known competitions such as the DARPA Grand Challenge (Thrun *et al.*, *forthcoming*), the Trinity College Fire-Fighting Home Robot Challenge (Verner 2004), the AAAI Robot competitions (Balch and Yanco 2002) and RoboCup (Kitano et al., 1997) have done much to advance the fields of robotics and artificial intelligence as well as the visibility of robotics research in the eyes of the media and the general public. However, each of the existing competitions has some major roadblocks for undergraduates wishing to participate. The costs involved for equipment purchase and maintenance, along with travel to competitions can make participation difficult for students attending most typical colleges.

The pedagogical benefits of robot competitions are numerous. As a discipline, robotics utilizes many of the general curricular components taught in most computer science curricular. Students need to be proficient in programming, software development, software engineering, configuration management, and networking. Knowledge of hardware integration and management is necessary as well. In terms of curricula in Artificial Intelligence, embodied robots give students practical experience in AI concepts such as machine learning. Fundraising and team budget management show the students the need for good business management skills. In order to build upon past years' experiences, students learn to recognize the usefulness of written documentation. Interpersonal skills that are not easily learned from an academic course, such as team leadership and organization, are gained as well.

Below we describe several of the better known college level competitions, along with some of their disadvantages with respect to extracurricular robotics. The *DARPA Grand Challenge* requires each competing team to build an vehicle that can autonomously navigate in a real-world environment. The equipment needed and the technical expertise for this competition is beyond what is available at most colleges and universities. The *RoboCup*[1] leagues present similar roadblocks. Probably the least expensive league in RoboCup soccer is the Four-Legged league, which is designed to keep costs to under US$10,000. This is still beyond the scope of many Computer Science department budgets, and certainly most club budgets. The simulation league, although affordable, does not offer the intrinsic motivation that an embodied architecture does. *FIRA*[2] has several divisions. The more introductory leagues are based on the Khepera robot. The basic Khepera with the K213 vision turret, lists for approximately US$4,000. Again, due to its expense, this platform is not a viable alternative for extracurricular robotics. Hardware costs vary in the *Trinity Firefighting Contest*, bounded by the dimensions specified in the rules of the competition (Verner 2004). Though Trinity College's event has various competitive levels, university teams must travel to Hartford, Connecticut, and funding for travel may not be subsidized by students' universities, particularly for groups of students to travel as teams, thus requiring them to somehow shoulder travel costs privately. Loosely organized contests, such as the KISS Institute's *Beyond Botball*[3], allows any adult beyond the high school years to compete. Similar to the Trinity competition, robots are built to contest specifications. Again the major cost, aside from the robot equipment, is to travel to the KISS sponsored National Conference on Educational Robotics. The purpose of the *AAAI Mobile Robot Competition* is to challenge the AI and robotics communities with increasingly difficult tasks so as to forward research in these areas (Balch and Yanco 2002). It has been a staple of the AAAI conference each year since 1992. Although undergraduate teams have participated in this event, most tasks are complicated, requiring extensive knowledge in these fields. Hence the degree of sophistication needed to effectively participate, precludes entry by most average undergraduate students.

---

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

[2] http://www.fira.net

[3] http://www.botball.org/season/2006/beyond_botball.php

Our Approach

Involving more university students in extracurricular robotics necessitates consideration of a robotic platform's cost versus its educational benefit, and availability of local competitions. There are several low-cost platforms that will suffice (Dodds 2006). At the College of Staten Island (CSI), the Computer Science Club was encouraged to purchase and experiment with the LEGO Mindstorms Robotics Invention System (RIS). The Mindstorms robot is built around a Hitachi microprocessor, embedded in a LEGO brick called the "RCX", and offers a sufficiently challenging and extendable platform for extracurricular robotics. The kit comes with a variety of sensors and a large number of LEGO pieces that can be configured into many imaginative robot body designs. Although the RIS comes with a graphical programming interface (either "RCX Code" or "RoboLab", depending on where the kit is purchased), several additional compilers, each based on different high-level computer languages, have been created for the RCX by hobbyists. As an on-going project, the club has been involved in building and competing with soccer robots using the RCX brick, to participate in RoboCup-inspired ELeague soccer games[1].

The CSI Computer Science Club

Maintaining a continued interest in a student club has many challenges, especially for a commuter school such as the College of Staten Island. Student's work, school and home life compete for time spent on club initiatives. According to (Gersting 1998), a university computer science club has several purposes: students share their computer related experiences and offers each other encouragement, arrange for guest speakers and field trips, provide opportunities for social activities, counsel each other about curriculum requirements, courses, and faculty, and do service for the school and department. In addition, computer science clubs can act as forums for students to expand on curricular topics. The CSI computer science club has a web server, giving students a chance to program, setup, and maintain a web site used by all student clubs. Members have also used the club as a way to expand upon their interest in robotics. The introduction of robotics into the CS1 and Artificial Intelligence classes piqued the interest of many club members. The desire to "play" with robots led to students becoming involved in organizing a low platform robot soccer team. Engaging in robot soccer allowed members to participate in a friendly competition, meet students from other universities, engineer better robots, program robots, and, in general, to have fun.

Organizing a Robot Soccer Team

One of the fundamental reasons for continued student involvement in CSI's computer science club stems from the support given by the Computer Science Department. Professors are more than willing to allow club members to speak to their classes about the benefits of club membership. Professors announce and encourage their students to participate in club activities, using robot soccer as an enticement for membership. The Computer Science department also provides the club with it's own office space, and the department and school administration also support club projects financially. CSI's Office of Information Technology has been generous, supplying the club with computers, printers, and several Mindstorms kits.

The investment by the department and administration is not without payback. By maintaining a web club server, the club performs a service to both other student clubs and the school. Club members have volunteered to attend department sponsored recruitment events. By demonstrating their robots and talking about projects and events, the club demonstrates the benefits of a computer science major to potential students. Student involvement in the club also helps with retention; students who join the club tend to finish their studies at CSI. Since the club office is located among computer science faculty offices, students tend to interact with faculty outside the classroom.

Robot soccer has become an ongoing project for CSI's computer science club. Unlike class projects, club projects take place over longer periods of time, since students participate in their "spare" time. This presents many logistical problems. The robot soccer project started in Fall 2004, and the project is still in its beta phase, even after several years of club involvement. Keeping a project viable over a long period of time, with changing club membership due to new members joining, older ones not joining or graduating, is difficult.

Several factors contribute to the ongoing interest in this project. Junior club members are encouraged to participate in the robot soccer project, allowing them to build and program club robots. Current club administrators personally encourage more junior members to run and organize events, robot soccer being one of them. This gives them confidence in their own organizational skills, with many of these junior members eventually becoming club administrators. In addition, robots are stored and displayed in the computer club office making them accessible to all members. Also, the club advisor's enthusiasm for the robotics project has much to do with its continuation.

Funding the Robot Soccer Team

Since ELeague regulations call for four robots on a team, the cost, just for a set of Mindstorms, is about US$800. Adding to that the need for computer and camera equipment, initial setup can run close to US$2000. To cover this cost, CSI's club obtained funds in several ways. Each club at CSI receives a budget of approximately $1000 per year to pay speakers, and to purchase supplies and refreshments for club meetings; there is not much available for robot purchases. Fundraising serves as another source of club income. CSI's club has started an annual alumni event. Alumni, current students and faculty are treated to a speaker and refreshments during this event. Participants are

---

[1] http://agents.sci.brooklyn.cuny.edu/eleague

gently encouraged to purchase raffles to help support the club's projects. School administrators have also been financially supportive. CSI's OIT department was able to use some of their funding to purchase the club several robotic kits and video equipment.

Technical Implementation

The notion of an undergraduate league for RoboCup was originally proposed in 2003 (Anderson et al., 2003), which was subsequently renamed to "entry-level" or *ELeague*. The goal of the ELeague is to provide an intermediate step from participation in RoboCupJunior[1] (RCJ) to participation in the Small-size or Mid-size leagues of RoboCup. A significant jump in both expertise and resources are required to be competitive in these leagues, as compared to the Junior league. It has been estimated that it takes at least two years to build a RoboCup team from scratch, which is a large time commitment for undergraduate students. In addition to the large difference in technical sophistication from RCJ to the senior leagues, the costs involved in the robotic equipment, as mentioned earlier, can be prohibitive.

The solution was to design a league that not only involves lower start-up costs but also addresses two of the major technical issues that tend to slow down or prevent undergraduate (or any entry-level) teams from participating at a competitive level: *vision* and *communication*. The approach is for the ELeague developers and event organizers to provide standard, low-cost hardware and software for these aspects. The vision hardware employs an X-10[2] surveillance camera connected to a Conexant[3]-based framegrabber card, e.g., a Hauppage[4] WinTV PCI board, via S-Video. The vision software runs Video4Linux[5] to talk to the framegrabber card, which interfaces to a software *vision server* for image processing. Communication is handled by buffering messages in a *communication server* and then sending them to robots using an infra-red (IR) transmitter. A strict message format is used such that each team client supplies a short string containing a command for one or more of their robots; the two messages are concatenated and a single message is broadcast. Each team's robots are programmed to receive the messages, decode their segment of the message and act according to the command(s) received.

The original software system design, detailed in (Anderson et al., 2003), called for a vision server to broadcast position information about robots and the ball directly to team controllers, which would then, in turn, send commands to a communication server. As described below, we have modified the system architecture for two reasons. First, the new architecture allows for easy control by one team, which makes development simpler to manage. Second, the new architecture uses network resources more efficiently for two team competition setups. An overview of the new system architecture is shown in figure 1.

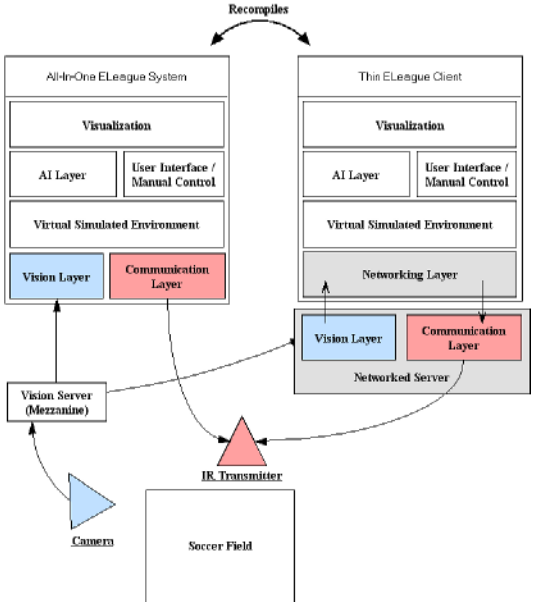


Figure 1. New System Architecture

The new system design takes a layered, object-oriented approach, providing unidirectional communication with robots and maintaining a simulation environment, all within an easy-to-use and install package. The system was developed and implemented under Linux and makes use of several cross-platform libraries. Windowing and font rendering is provided by the Simple Directmedia Layer[6], and rasterization is provided by OpenGL[7]. The system can be built in two modes. The *All-in-One Engine* mode provides operation of the entire system as one process and is designed to use for standalone development. The *Thin Eleague Client* mode is designed for multi-team development and competition. This thin client version includes a *Networked Server Layer*, which is responsible for reading position information from the vision layer, sending it to each team, receiving robot commands from each team and transmitting commands to robots using an IR transmitter.

The *Vision Layer* includes a camera and vision server software which writes position data of robots and the soccer ball to a shared memory buffer. Initially, the *Doraemon*[8] video server package (Baltes, 2002) was used, but currently *Mezzanine*[9] is employed because it is easier to

---

[1] http://www.robocupjunior.org
[2] http://www.x10.com
[3] http://www.conexant.com
[4] http://www.hauppauge.com
[5] http://linuxtv.org/v4lwiki/index.php/Main_Page
[6] http://www.libsdl.org
[7] http://www.opengl.org
[8] http://sourceforge.net/projects/robocup-video
[9] http://playerstage.sourceforge.net/mezzanine/mezzanine.html

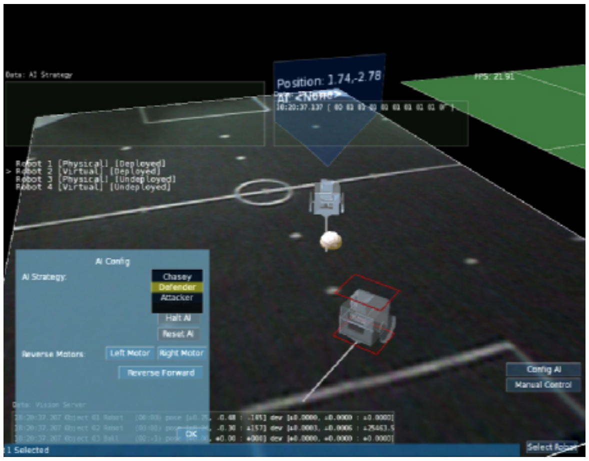
install and it handles lens distortion more robustly, which is a necessity with inexpensive cameras such as the X-10. An adapter interface is used to abstract away the details of communicating with the Mezzanine server, which allows any vision server to be integrated painlessly by the creation of an appropriate adapter class.

Robots receive commands from an IR transmitter connected to the computer by a serial or USB port. Such interfaces are generally messy and platform-specific in their implementation. In the new solution, a “communications object” in the *Communication Layer* hides the details of any IR-transmitter-to-robot transmission-encoding format and the platform specifics of serial/USB port communication. This object reads the robot states from the virtual environment being maintained by the solution, and sends commands to all of the robots on the field; effectively eliminating all low-level I/O concerns, allowing the programmer to concentrate on the AI logic by manipulating high level objects.

Each team runs its own *AI Layer*, a controller that determines what robots should do based on state information received from the *Vision Layer* and any internal state information stored by the team. The AI Layer functions on a simulated environment that is filled with objects such as robots and balls. These objects are periodically synchronized with the soccer field visible to the vision server and can be optionally interpolated in between those synchronizations to provide smoother data. They are manipulated by the programmer using each team’s own implementation of “AI-Strategy” interfaces. These interfaces in turn manipulate more low-level objects which the robots on the field are periodically synchronized against. These lower-level synchronizations are transparent to the programmer. An intermediate set of data structures allow the AI Layer to remain ignorant of the hardware-dependent input and output layers.

To aid in the rapid development of capable robots, the AI Layer comes with a namespace for sharing useful behaviors between different AI strategies such as Goalies and Attackers. For example, the shared behavior “DriveToTarget” drives a robot to a target by moving and turning. In a group environment, having a shared repository of basic behaviors as the basis of an AI framework helps new teams (or team members) get a quick start and is useful for rapid testing of new ideas.

In addition, new a *User Interface* (Figure 2) and *Manual Control Layer* and a *Visualization Layer* have been included that can be instantiated and kept up-to-date by the vision layer, allowing control of virtual robots on a soccer field. This lets students test and debug robot control algorithms without needing the complete soccer pitch setup. The system is also handy for development, as well as calibration and tuning for “away” games. In the visualization window, a soccer field is rendered on the screen, and on it are drawn any robots present in the game. Virtual robots that have no physical counterparts and exist solely for entertainment or debugging purposes are semi-transparent, as if they are ghosts. Real robots are drawn as completely opaque. Both are represented using 3D models, which are oriented and positioned on the virtual pitch. The display is overlaid with debugging information, such as the data being sent over the communication ports, the data coming in from the vision server, and anything the AI strategies wish to tell the programmer. As well, an overlay is drawn for any selected robot. This overlay may display debugging information such as the positions of objects relevant to the robot, or the path it is planning to take to reach its target—all in the form of 3D shapes and floating text bubbles.



```
  Robot 1 [Physical] [Deployed]
> Robot 2 [Virtual] [Deployed]
  Robot 3 [Physical] [Undeployed]
  Robot 4 [Virtual] [Undeployed]
```

Figure 2. Screenshot of interface

The *Manual Control Layer* allows robots to be controlled by a user. Properties may be adjusted, such as the repairing of a motor that is spinning the wrong way. There is room here to create more adjustments, such as correcting different motor speeds without having to hardcode such robot-specific adjustments into the AI. It is possible to have virtual robots playing with the real robots within the simulation layer by manipulating the virtual robots using the keyboard and mouse, or by assigning them a pre-defined AI-Strategy. This strategy will run purely in simulation, without being affected by the data coming in from the vision server. Real robots may be affected by the actions of virtual robots if their AI strategies choose not to ignore robots marked as “Virtual.”

Discussion

Our goal is to create an organized league where the bar is high enough to be challenging for undergraduates but setup is easy and inexpensive. Once we have an organized league, we need to know what defines whether or not we have been successful. We know competitions like those mentioned earlier are successful since, over the years, the number of teams participating in these has grown. We would like to see the same type of growth with the ELeague. More faculty need to be involved, both as team advisors and in an organizational capacities for the league.

We can measure our future success quantitatively and qualitatively. For example, we can count the number of students that go on to study robotics at the graduate level. Attitudinal surveys can assess students' perceptions towards robotics, artificial intelligence and computer science in general. Students can be surveyed as to whether they are interested in pursuing graduate degrees, and or undergraduate/graduate research. We will look to earlier work evaluating robotics competitions to help with these measures (Sklar, Eguchi and Johnson 2002; Verner 1997; Verner 1998).

Conclusion

In this paper, we describe our long-term goal of establishing of an undergraduate robotic soccer league. We present a new formulation of the RoboCup-inspired ELeague as a means to achieve this goal. An extracurricular approach to setting up such a league is appealing and obtainable since many schools have extracurricular clubs and computer science organizations. Given new developments in technology and the emergence of inexpensive robots such as LEGO NXT, as well as lower cost technological enhancements such as off-the-shelf vision software and sonar, the current configuration of the ELeague is designed to be able to grow and incorporate enhancements such as these. For the future, we hope that more of our university colleagues will become involved in the ELeague. Possible seminars, workshops and symposia can be held to help create a more widespread community.

Acknowledgements

Thanks to Matthew Ganz for his work with the College of Staten Island's Robot Soccer team and his contribution.

References

Anderson, J. Baltes, J., Livingston, D., Sklar, E. and Tower, J., (2003), Toward an Undergraduate League for RoboCup. In *Proceedings of the Seventh International RoboCup Symposium (RoboCup-2003).*

Balch, T. and H. Yanco (2002), Ten years of the AAAI mobile robot competition and exhibition: looking back and to the future, *AI Magazine 23:1*, pp.13-22.

Baltes, J. (2002), Doraemon: Object orientation and id without additional markers. In *2nd IFAC Conf on Mechatronic Systems*. Amer. Automatic Control Council.

Dodds, Z., Greenwald, L., Ayanna, H., Tejada, S., and J. Weinberg (2006), Components, Curriculum, and Community: Robots and Robotics in Undergraduate AI Education, *AI Magazine 27:1*, pp 11-22.

Dodds, Z., Santana, S., Erickson, B., Wnuk, K., Fischer, J., and M. Livianu (2004), Teaching Robot Localization with the Evolution ER1, *AAAI Spring Symposium.*

Fagin, B.S., Merkle, L.D., and T. Eggers (2001) Teaching Computer Science With Robotics Using Ada/Mindstorms 2.0, Proceedings of *ACM SIGAda Int'l Conf on Ada.*

Gersting, A., and F. Young (1998), Service Learning Through the Computer Science Club, *ACM SIGCSE Bulletin 30:4*, pp 25-26.

Greenwald, L., and D. Artz (2004), Teaching Artificial Intelligence with Low-Cost Robots, *AAAI Spring Symposium.*

Imberman, S. (2004), A Laboratory Exercise Using LEGO Handy Board Robots to Demonstrate Neural Networks in an Artificial Intelligence Class, *AAAI Spring Symposium.*

Imberman, S. and R. Klibaner (2005), A Robotics Lab for CS1, *21st Annual Consortium for Computing Sciences in Colleges Eastern Conference*, October 2005.

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

Parsons, S., and E. Sklar (2004), Teaching AI using LEGO Mindstorms, *AAAI Spring Symposium.*

Sklar, E., Eguchi, A., and J. Johnson (2002), RoboCupJunior: learning with educational robotics, In *Proc of RoboCup-2002: Robot Soccer World Cup VI.*

Sklar, E., Parsons S., and P. Stone (2004), Using RoboCup in university-level computer science education, In *Journal on Educational Resources in Computing (JERIC)*, 4:2 (June 2004) Special issue on robotics in undergraduate education. Part 1.

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 P. Mahoney (*forthcoming*), Stanley, the robot that won the DARPA grand challenge, *Journal of Field Robotics.*

Verner, I.M. (1997), The Value of Project-Based Education in Robotics. In *RoboCup-97: Robot Soccer World Cup I, LNAI 1395*, Springer Verlag.

Verner, I.M. (1998), The Survey of RoboCup '98: Who, How and Why. In *RoboCup-98: Robot Soccer World Cup II, LNAI 1604*, Springer Verlag.

Verner, I., and D. Ahlgren (2004), Robot Contest as a Laboratory for Experiential Engineering Education, *Journal of Educational Resources in Education, 4:2.*