AC 2009-1425: VIRTUALIZING FIRST FOR IMPROVED RECRUITMENT OF STUDENTS IN COMPUTER SCIENCE AND ENGINEERING

John Bowles, University of South Carolina

John Bowles is an Associate Professor in the Computer Science and Engineering Department at the University of South Carolina where he teaches and does research in reliable system design. Previously he was employed by NCR Corporation and Bell Laboratories. He has a BS in Engineering Science from the University of Virginia, an MS in Applied Mathematics from the University of Michigan, and a Ph.D. in Computer Science from Rutgers University.

Caitlin Buchhaults, University of South Carolina

Caitlin Buckhaults is an undergraduate student majoring in Computer Engineering at the University of South Carolina. She was the lead programmer for her high school FIRST Robotics Competition team at Blythewood (SC) High School and is currently vice president of the GamecockFIRSTers, a student organization that mentors high school students on the USC FIRST Robotics incubator team (Number 2815). During Summer 2008 she worked on a Research Experiences for Undergraduates project to adapt Microsoft Robotics Developer Studio to the FIRST Robotics Competition.

Donald Griffith, University of South Carolina

Donn Griffith is the Outreach, Recruitment, Retention Director for the College of Engineering and Computing at the University of South Carolina. Previously, he worked as a manager, team coordinator, and teacher with FIRST Robotics, FIRST VEX Robotics, FIRST Lego League, and Project Lead the Way, and as a high school industrial arts and technology teacher. He has a BS in Industrial Arts and a MEd in Industrial Education both from California University of Pennsylvania and an EdD in Career/Technology Education from Clemson University.

Virtualizing *FIRST* for Improved Recruitment of Students for Computer Science and Engineering

Abstract

The *FIRST* Robotics Competition is an international program aimed at inspiring high school students to pursue careers in science and technology through a yearly challenge in which teams of students design and build robots to compete against other teams from all over the world. The teams engage in an intense six-week effort to build a robot to do a specified task such as to scoop up a bunch of beach balls and put them into a large container. The robots are large and must operate both autonomously and with radio control. The contest itself has the atmosphere of a professional wrestling tournament, complete with a master of ceremonies and teams and cheerleaders in costume and often with painted faces and hair. All this has proven to be an effective draw for recruiting students to engineering and to a lesser extent to computer science. The College of Engineering and Computing at the University of South Carolina (USC) is sponsoring an unusual team composed of students from several high schools. Over time, and as interest grows to a sustainable level, the USC team will spin off teams to the participating high schools, further increasing the pool of students interested in science and technology.

One recurring problem is that the robot software cannot be tested and debugged until after the mechanical and electrical subsystems are functional and nearly complete; hence, the programmers have minimal time, at the end of the development period, to work with the robot and they encounter great difficulty and frustration in developing and optimizing their code. Implementing the robot and the contest environment in a simulator such as Microsoft Robotics Developer Studio allows the students to develop their software and hardware concurrently. This greatly improves individual team performance in the competition and provides students with increased exposure to software design, which should result in increased numbers of students entering into computing careers.

1. Introduction

Each year in January, the *FIRST* (For Inspiration and Recognition of Science and Technology) organization releases the rules and specifications for its annual Robotics Competition. Teams of high school students have six weeks from the time the challenge is issued until the finished robot must be shipped to the contest site. While the competition is to build a functional and competitive robot, *FIRST*'s main objective is to inspire high school students to pursue careers in science and technology—the robotics competition is simply the vehicle for achieving this goal. In this regard *FIRST* has been extraordinarily successful. Studies show, see Table 1, that *FIRST* alumni major in engineering at about seven times the rate of other high school graduates and they major in computer science at about twice the rate of other high school graduates[2].

Although having students major in computer science at twice the average rate of high school graduates in general is an excellent result, it pales in comparison to the seven-fold increase in engineering majors. We believe that a large part of the reason for this difference is a result of the manner in which the robot is developed. A team begins by developing the operational concept

Table 1. Comparison of FIRST Alumni to General High School Graduates in Educational Choices[2]

	Percent of FIRST Alumni	Percent of High School Graduates
Attend College	89%	65%
Major in STEM disciplines	55%	28%
Major in Engineering	41%	6%
Major in Computer Science	11%	5%
Plan on obtaining graduate degree	78%	60%

for their robot to solve the problem presented in the contest challenge. The "mechanical team" then designs and builds the mechanical superstructure for the robot after which the "electrical team" builds and incorporates the electrical subsystem into the robot. The greatest amount of time is dedicated to building the mechanical and electrical subsystems. Since the software cannot be tested and debugged until these subsystems are functional and nearly complete, the team programmers are left with only minimal time with the robot and thus encounter great difficulty and frustration in developing and optimizing their code. Building the contest environment in a simulator such as Microsoft Robotics Developer Studio and then implementing the robot virtually in that environment would allow students to develop their software and hardware concurrently. This would greatly improve individual team performance in the competition, provide students with increased exposure to software design, and encourage more students to seriously consider computer science as an attractive career choice.

To this end, we began using Microsoft Robotics Developer Studio (MRDS)[5, 6] to construct the tools that would enable high school students to build a simulated robot for the *FIRST* Robotics Competition during the summer, 2008. We are presently using these tools with the University of South Carolina (USC) Robotics team, Los Pollos Locos, team composed of high school students from four nearby high schools. The remainder of this paper describes the *FIRST* Robotics competition in Section 2, Microsoft Robotics Developer Studio in Section 3, and how MRDS complements and enhances *FIRST* in Section 4. Our plan to use the USC team as an incubator to grow additional high school teams is discussed in Section 5. Conclusions and our results to date are in Section 6.

2. The *FIRST* Robotics Competition

The *FIRST* Robotics Competition is run with the intensity of a professional wrestling tournament, complete with a master of ceremonies, loud music, and teams and cheerleaders in costume, often with painted faces and hair. The robots are large and operate both autonomously and with radio control. While the competition is intense and the teams work hard to accomplish their objectives, they also must adhere to an ethic of "gracious professionalism" and good sportsmanship.

For the 2009 competition the game is named "Lunacy" in recognition of the 40th anniversary of Neil Armstrong's landing on the moon. It features low traction flooring (and wheels) to simulate the low-gravity environment present on the lunar landscape. Alliances of three robots compete

to throw spherical 'Lunar Cells' into trailers hitched to three opposing robots while also protecting their own trailers[4]. The 2008 contest featured a race track on which robots drove counterclockwise and manipulated large, 10 pound balls over and under 6½ foot barriers in order to obtain points.

The 6 weeks from the time the game design is announced until the robot must be shipped are typically organized into 1 week for system design, 2 to 3 weeks for hardware construction, 2 to 3 weeks for electrical systems construction, and the remaining time for programming, testing, debugging, and practice. The teams then go to the local regional event, where they spend 3 days competing for the right to go to the championship event and for many other prestigious awards, including Safety, Gracious Professionalism, All Star Rookie, Team Spirit, and the coveted Chairman's Award which recognizes a team that exemplifies the values of *FIRST*..

As in the "real world", *FIRST* teams never have enough time to accomplish all of their goals. They have to work under high-stress situations in which a major setback can be devastating to the team's success. However, because the programming aspect of the robot's development depends on all the other aspects of the robot being completed first, even minor setbacks affect not only the systems in which they occur, but also the programmers' ability to effectively program the robot.

Computer Science as a field has a very high learning curve; the first programming language is often the most difficult to learn because students must learn logic and how to organize their code as well as the syntax and semantics of the language. *FIRST* students not only need to learn how to program—this is often their first significant programming experience—but they also must learn to control a complex system containing sensors, end-effectors, and pneumatic subsystems, all contained on a mobile robot that must be able to both respond to the operator's control and operate autonomously.

As noted above, this method of outreach, challenge, and inspiration has proven to be successful with measurable increases in the numbers of students entering related disciplines in college. In addition, the percentages of minorities entering into engineering reflect the percentages of *FIRST* alumni entering engineering as well. Women are more than 16 times as likely to enter into engineering (33% versus the national average of 2%) , African Americans are more than 5 times as likely to enter into engineering (27% versus 5%), and Hispanics are more than 7 times as likely to enter into engineering (47% versus 6%).

3. Microsoft Robotics Developer Studio

Microsoft recently introduced a robotic simulator named Microsoft Robotics Developer Studio (MRDS)[5, 6]. MRDS allows individuals to build complex robotic and environmental systems in the absence of a physical robot. It is currently free to download and is being used in several other robotic competitions; this can be a major asset to *FIRST* Robotics Competition teams.

MRDS includes the Visual Programming Language (VPL) as part of the simulator package. VPL is similar to Lab View, a commonly used programming tool in both *FIRST* and in industry. It utilizes a graphical user interface with connected boxes representing logic statements and

robotic entities. This gives students the control and much of the logic of standard industry languages, but makes programming easier by having a visual user interface instead of command line text.

The MRDS physics engine is very advanced. It gives programmers the ability to specify the robot environment in great detail; for example, parameters control the bounciness (elasticity) of the floor, the weight distribution on the robot, and white noise in the air that can interfere with the robotic sensors. It also automatically generates default values for attributes of the robot and the environment that the user does not specify, thereby allowing the programmer to control as much, or as little, of the simulated world as needed. This allows programming teams to be very specific about the parts of the environment that are particularly important to game play, while not wasting time with extreme detail in the parts that are less critical.

One of the *FIRST* Robotic Competition's awards is based on teams' drawings of their robots in Autodesk Inventor[3]. Many teams compete for this prize, or simply create an Autodesk virtual copy of their robot for their own purposes. MRDS allows students to import these drawings into its simulator to create a very detailed physical simulation model of the robot. For example, robots with ridges on the wheels will appear to drive slightly less smoothly than robots with smooth wheels, and they will appear to have better traction in the simulation environment.

MRDS can be programmed in many languages, including (but not limited to) C#, C++, Visual Basic, Python, and its own VPL. Students who already know one of these programming languages can take advantage of their prior learning and save time by programming in that language instead of having to learn a new language.

Microsoft Robotics Developer Studio is also in use in several other robot competitions. The Robocup, a yearly event hosted in Suzhou, China, is a series of competitions aimed at creating a robotic soccer team to beat the 2050 world champion soccer team. The competitions include individual robots ranging in size from smaller than six inches to human sized robots[8]. One competition is based entirely within the simulation environment and uses MRDS to run the game matches and determine team wins and losses. Microsoft also has its own yearly competition in which teams compete with iRobot Creates in a sumo match[5].

The Princeton Autonomous Vehicle Engineering (PAVE) group competes in the yearly Defense Advanced Research Projects Agency (DARPA) Grand Challenge[7]. In the DARPA Challenge, teams build an autonomous car that navigates rough terrain or urban traffic and compete based on overall performance, durability, and safety. PAVE has used MRDS to simulate troublesome environments before risking their autonomous car in those risky situations.

4. *FIRST* and MRDS

Since *FIRST* teams have limited time, and every delay can decrease the already minimal amount of time that programmers have with the robot, using a simulator to work on the robot's code while the robot is being built can be a great asset to the overall performance of the team. Programming teams using Microsoft Robotics Developer Studio would have five weeks to work on the robot's software (the six-week develop period, less one week for planning strategy)

instead of waiting for four to five weeks while the mechanical and electrical teams create the physical robot. The programming team would then need only a few days to work on accounting for differences between the simulated and real world physics (such as motor bias in the robot) instead of working on code that, for instance, would allow the driver to operate the pneumatic subsystem.

This also means that any physical setbacks during the construction of the robot are less of a crisis for the software team. Additionally, more students are exposed to programming, since such exposure is not hampered by construction bottlenecks.

In the Lunacy challenge for 2009, the floor consists of a gel coated, fiberglass reinforced polymer material that would cost approximately $1,000[1] to replicate for practicing purposes; this is not a realistic option for most budget-constrained teams. By simulating this low-friction environment teams can test their robot's capabilities and practice handling it under these abnormal circumstances while the robot is being built and they are still able to modify the robot to minimize any weaknesses discovered.

The student driving team can use Microsoft Robotics Developer Studio to practice competitive driving in the simulated environment while the robot is being built and even after it has been shipped. They can even use joystick control for the simulated robot due to the compatibility of MRDS with most external controllers. (MRDS has been successfully tested with driving simulated robots under the guidance of an Xbox 360 controller.)

One of the major difficulties in error testing is isolating the problem and determining what to fix when the robot behaves unexpectedly. Students can spend hours trying to 'fix' correct code, or tracing wires that are connected properly. With Microsoft Robotics Developer Studio, however, running a simulation of the robot can often help localize the problem as being either a physical problem or a software problem. Simulation can also allow students to discover design flaws and modify the robot's design before the flaws are implemented. This saves both time and money by minimizing the number of times that the physical robot must be modified.

Unfortunately, the current *FIRST* control system and Microsoft Robotics Developer Studio utilize different programming languages. Some of them are similar, such as C (*FIRST*) versus C# (MRDS) and Lab view (*FIRST*) versus VPL (MRDS); however, future versions of MRDS may be able to bridge that gap and make MRDS an even more effective tool for the *FIRST* Robotics Competition.

5. Incubating Teams

The College of Engineering team presently has 16 student members drawn from four local high schools. The College, through a student organization, provides space and shop facilities to build the robot, college students who mentor the high school students, and organization to get the job done. Many of the college students participated on their own *FIRST* teams when in high school. Parents, teachers from the participating high schools, and industry professionals also mentor the high school students.

Since its founding in 1989, *FIRST* has grown to over 1,680 teams with an estimated 42,000 students for the 2009 competition. There are 300 rookie teams, including the USC team, in the 2009 mix. Most teams have between 15 and 30 members and the Robotics teams are fed by alumni of the *FIRST* LEGO League for 9 to 14 year-olds and the Junior *FIRST* LEGO League for 6 to 9 year-olds. Each rookie team is mentored by an experienced team which provides a resource for solving both the technical and managerial problems that inevitably arise in the first year of competition and eases the learning curve of participating in the competition. Over time, and as the number of participants from individual high schools reaches a sustainable level, the USC team expects to spin off teams to each high school.

6. Conclusions

We are presently using MRDS to develop a simulated robot for Lunacy for the robotics team at the University of South Carolina. The team benefits from implementing their robot in Microsoft Robotics Developer Studio in several ways with the result that they will perform better in the competition. When applied more broadly, this will result in fewer teams leaving the competition due to their inability to succeed. Longer term, more people will become *FIRST* members, alumni, mentors, coaches, and sponsors; and more students will become interested in careers in science and technology.

Finally, through increased numbers of *FIRST* students programming robotic systems more successfully, more people can be expected to pursue careers in Computer Science (and realize a recruiting potential similar to that in engineering). These students will not only enter Computer Science with real-world programming experience, including severe time and budget constraints, but, because of that knowledge, they will be less likely to drop out of Computer Science and a computing career because of the steep learning curve. More students completing degrees in Computer Science will mean that the three jobs for every graduate ratio that currently exists can begin to decrease and our country can better maintain its technological edge in the world.

Implementing Microsoft Developer Studio to simulate robots in the *FIRST* Robotics Competition will positively impact *FIRST* and the Computer Science field.

7. Acknowledgements

This work was done at the University of South Carolina as part of the Research Experiences for Undergraduates in Multidisciplinary Computing project and supported in part by the National Science Foundation (Award # 0649105) and the STARS Alliance, National Science Foundation (Award # 0739216).

Bibliography

1. ".900 IN. FRP Wall Panel 4 Ft. by 8 Ft. White."< http://www.homedepot.com/webapp/wcs/stores/servlet/ProductDisplay?storeId=10051&langId=-1&catalogId=10053&productId=100389836>

2. Alan Melchior, Faye Cohen, Tracy Cutter, and Thomas Leavitt, “More than Robots: An Evaluation of the *FIRST* Robotics Competition Participant and Institutional Impacts”, Brandeis University, Center for Youth and Communities, Heller School for Social Policy and Management, Waltham, MA, April 2005. <http://usfirst.org/uploadedFiles/Who/Impact/Brandeis_Studies/FRC_eval_finalrpt.pdf. >

3. "Autodesk - Autodesk Inventor - Autodesk’s Ongoing Commitment to *FIRST* Robotics Competition Nurtures Students’ Love of Science and Technology." Autodesk - 2D and 3D Design Software for the Architecture, Engineering, Construction, Manufacturing, and Entertainment Industries. Autodesk, Inc. <http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=4243891&linkID=8391941>.

4. *FIRST* Robotics Competition 2009 Season Kickoff. NASA Robotics - Events: *FIRST*. 3 Jan. 2009. http://robotics.nasa.gov/events/2009_frcwebcasts.php

5. Johns, Kyle, and Taylor Trevor. *Professional Microsoft Robotics Developer Studio*. Indianapolis: Wiley Publishing, Inc., 2008.

6. Microsoft Robotics Studio. Computer software. Ver. 1.5. Microsoft Robotics Studio 1.5 Refresh. 26 Dec. 2007. Summer 2008 <http://www.microsoft.com/downloads/details.aspx?familyid=73092FF6-E37B-45C6-8E5E-C23D5D632B1E&displaylang=en>.

7. "Princeton Autonomous Vehicle Engineering". Princeton University. <http://pave.princeton.edu/main/>.

8. "RoboCup Official Site". The RoboCup Federation. <http://www.robocup.org/02.html>.