INVESTIGATING OPEN SOURCE SOFTWARE AND EDUCATIONAL ROBOTICS

Keith J. O'Hara and Jennifer S. Kay
Computer Science Department
Rowan University
Glassboro, NJ 08028
{ohara, kay}@elvis.rowan.edu

ABSTRACT

We investigate the role of open source software in the learning, teaching, utilization, and development of educational robotics and its corresponding core technologies. First, we present a survey of current open source solutions for the LEGO® MindStorms™ Robotics Invention System™. Next, we discuss the role of open source software in the development of two undergraduate robotics projects.

INTRODUCTION

Educators are using robots increasingly across the curriculum. Using robots in the classroom at the primary school level promotes team problem solving skills and interest in science [12]; at the university level they reinforce computer science and engineering concepts [4, 5, 6, 8, 10, 14]. Robots have also been used to teach non-majors computer science, for instance, introducing programming to teachers pursuing a degree in Educational Technology. [11] Robotics is a flexible and powerful channel to demonstrate a variety of concepts and demands flexible and robust tools. In an educational setting, however, these tools must also be inexpensive. We must acquire, use, develop, and distribute, tools for educational robotics in a robust, flexible, and inexpensive manner. Robotics has also proven to be a successful educational device for encouraging cooperative problem solving and team work in computer science. [4, 6] This teamwork can be expanded beyond the classroom to include much larger projects and distributed teams.

Open Source Software (OSS) can help us use robotics to teach, learn, and apply computer science. In this paper, we discuss the role of open source software in educational robotics. First, we present several open source solutions for the LEGO® MindStorms™ Robotics Invention System™. Next, we discuss the role of open source software in the development of two undergraduate robotics projects, the Mouse Positioning System and RCX-NLI. The Mouse Positioning System uses computer mice as inexpensive positioning devices for

mobile robots. RCX-NLI provides a natural language interface to a LEGO MindStorms robot. Both were developed using an array of open source software.

OPEN SOURCE SOFTWARE AND EDUCATION

The Open Source Initiative's definition of open source software is: "software that must be distributed under a license that guarantees the right to read, redistribute, modify, and use the software freely." [31] As much a part of the Open Source Software (OSS) development model as the open source code is a global network of developers. The software goes through a type of natural evolution - resulting in rapid development, increased reliability, and decreased cost.

Probably the most popular piece of software to come out of the OSS movement is the GNU/Linux operating system. Linux can be found in a variety of applications beyond the workstation, from humanoid robots [19] to the TiVo® digital television recorder [13]. The reliability, quality, and affordability inherited from the OSS model, coupled with the ability to adjust its complexity according to its application makes Linux an excellent candidate for education. Linux is unique in that it is a candidate for both the robot's software system as well as the developer's workstation! It is this type of flexibility that makes Linux, and OSS in general, an excellent candidate for educational robotics.

Open source software can provide us with free or lower-cost technology in the classroom that we might otherwise be unable to afford. Also, by using and developing open source software, computer science students can participate in a large distributed software development community. They will interact with large "real-world" code-bases and will partake in the all-important process of peer-review.

OSS AND EDUCATIONAL ROBOTICS - A SURVEY

LEGO has produced a relatively inexpensive robotics kit, the LEGO MindStorms Robotics Invention System. The MindStorms system consists of a variety of traditional LEGO parts, as well as sensors, motors, and the programmable RCX "brick", which are designed to interlock with the standard LEGO pieces. These different components are combined to build a robot platform.

The RCX "brick" is a Hitachi H8 microcontroller with 16K of ROM and 32K of RAM. Programs written on a PC can be downloaded via an infrared (IR) link to the RCX. The programs rely on RAM resident firmware and a "System ROM" to run routines. [3]

While LEGO packages the MindStorms kit with a visual programming environment, many students may find themselves using C [21], Ada [5], or even Prolog[9] to program their RCX brick.

Much of the success of the LEGO RCX can be attributed to a hard working community that produces many of these alternative programming solutions, most of which are open source projects. We now present a survey of the current open source programming solutions for the LEGO RCX. More detailed descriptions of the different programming solutions can be found in [3, 7, 11].

The solutions vary in their choice of open source license. This choice is often a balance between how much of the code the developer wants to assure will

remain open and how wide-spread the software will be used. The two extremes of this spectrum are the GNU Public License (GPL) and the Berkeley Software Development (BSD) License. The GPL assures all future users will have the right to read, modify and redistribute the source code via a copy-left mechanism. This copy-left mechanism maintains that any derived works based on the GPL source code will also be released under the GPL. In contrast, the BSD license permits the source code to be used in any (possibly closed source) projects. Licenses like the Mozilla Public License (MPL) aim to strike a balance between these two extremes, by making the "core" code copylefted, but allowing the addition of proprietary features. Another option is to release the code under multiple licenses.

As shown in figure 1, the software can be divided into 3 categories. The first category of solutions use the standard RCX firmware. This allows users to easily switch back and forth between the alternative solutions and the LEGO visual programming environment. Programs are compiled on the host to the byte-code level and then transferred to the RCX to be interpreted by the byte-code interpreter.

The second category of programming solutions requires special firmware to be installed on the RCX. This permits more flexibility in programming the RCX, but also prohibits the use of other types of programming, such as the LEGO visual programming environment during the time this application specific firmware is installed.

The final category of software controls the RCX from the workstation. The MindStorms system comes with an IR transmitter/receiver that is used to down-load software to the RCX. However, it is also possible to leave the bulk of the computation on a workstation and transmit commands to the RCX via the IR link.

Contrary to some expectations, LEGO has not tried to discourage third party software for its MindStorms systems. Initially, LEGO ignored the open source community, but they have become increasingly supportive. In fact, during the development of the Robotics Invention System (RIS) 2.0, they released the software development kit prior to the actual release of the product. [3] This is a wonderful demonstration of how for-profit companies can actually benefit from open source software. The OSS solutions listed in figure 1 actually increased the value of the RCX hardware and contributed to its success. These projects enable students to use higher level programming languages, rather than LEGO's visual programming environment, appropriating its use at the university level. [5]

CASE STUDIES

We have experienced the benefits of open source software in computer science education by using it together with the LEGO MindStorms kit in two undergraduate robotics projects: The Mouse Positioning system and RCX-Natural Language Interface.

The Mouse Positioning System (MPS) began as an undergraduate research project in mobile robot localization and educational robotics. The system uses computer mice as inexpensive positioning devices for mobile robots. Its goal is to devise a positioning system that can be used in high school and undergraduate robotics project in a simple manner. The MPS was prototyped using a LEGO

Name	Programming Language (Dialect)	Additional Software Required	License
Standard Firmware			
Not Quite C [27]	C		MPL
Ada [5]	Ada	NQC, AdaGIDE, GNU AdaTranslator	Source Available[1]
Lego/Scheme [22]	Scheme	Lego MindControl	Special (BSD-Like)
Specific Firmware			
Librcx [20] (low-level)	C	GNU HC8 Cross-Compiler	MPL
LegOS [3, 21]	C	GNU HC8 Cross-Compiler	MPL
LeJOS [23]	Java	Java JDK 1.2	MPL
TinyVM [38]	Java	Java JDK 1.2	MPL
PbFORTH [3, 32]	Forth		Source Available
Controlling the RCX From Workstation			
Legolog [9, 24]	Golog/Prolog	NQC, (SWI, ECLiPSe, or LPA) Prolog	Special (BSD-like)
Pylnp [33]	Python	legOS, Python	Source Available
Lego::RCX:pm [25]	Perl	Perl	GPL/Artistic
Lego MindControl[26]	Visual Basic	Spirit.OCX	Source Available
RCX Java API [34]	Java	Java JDK 1.1 and Java Communications API	LGPL
RCXPort [35]	Java	Java JDK 1.1 and Java Communications API	GPL

Figure 1: Summary of popular OSS for the LEGO RCX

MindStorms mobile robot, Logitech optical and standard mice, and a C++ software system.

The MPS benefited greatly from the use of open source software. The general-purpose mouse (GPM) software, developed under the GPL, became invaluable, acting as a computer mouse tutorial. It laid out the technical details of different mice protocols.

The system also benefited from the use of open source development tools: Linux as the underlying platform, open source compilers (g++), and other tools (GNUEmacs, gdb). The prototype system used a LEGO MindStorms robot that

1The source is publicly available on the web, but not explicitly released under any license.

was programmed using Not Quite C. Once the system was developed we used an open source GUI toolkit, the fast light toolkit (FLTK), to begin writing demo applications. [17] One such demo application tracked the robot's movements on a simple grid (figure 2). In a little more than one semester, a single undergraduate was able to go from concept to the prototype system. The speed of this development cycle was clearly accelerated by the use of OSS.

Because of the standards compliance and flexibility of the Linux operating system, the software is very portable. This portability will be capitalized on as we move the Mouse Positioning System to other hardware and software configurations. The benefits are beginning to be observed in early trials where some of the FLTK-based demo applications, initially designed for a workstation, are now being run on a Linux powered handheld computer manufactured by Agenda Computing, the VR3. [16] Porting the software can be as easy as resizing the GUI and cross-compiling the source. NQC was also ported to the VR3 with changes only to its Makefile. [28] Finally, since the code is open and freely available, students from other schools have been able to benefit from the research.

RCX-NLI, a natural language interface to a LEGO MindStorms robot was developed as a semester project in an artificial intelligence class. It utilized Not Quite C and SWI-Prolog to accept simple commands and queries to control the robot via a text-driven interface (see figure 3). The natural language interface could be easily extended to use speech recognition software, e.g. IBM's ViaVoice, to have spoken language capability. Another possible modification would be to use speech-to-text software such as the open source speech synthesis engine festival. [18] The natural language interface was written in Prolog. It was inspired by the Legolog system developed by the University of Toronto's Cognitive Robotics Group. [24] Legolog is a Golog implementation for the LEGO MindStorms in which the RCX is controlled by the host computer running the Golog planner.

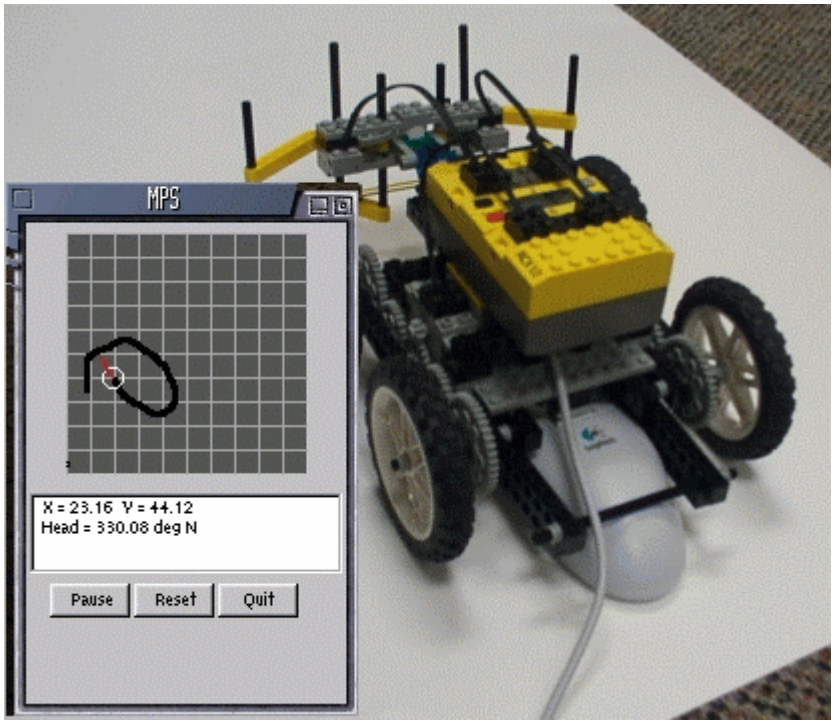

Figure 2: The Mouse Positioning System prototype. A screen shot of a demo program modified to run on the Linux powered handheld computer, the VR3, that tracks the robot's movements.

- **robot prompt> ?-** Please move forward.
  success.
- **robot prompt> ?-** Back up.
  success.
- **robot prompt> ?-** Go forward for 3 seconds and then turn on autopilot for 5 minutes.
  success.
- **robot prompt> ?-** Back up.
  success.
- **robot prompt> ?-** Turn left for 2 seconds and then go forward.
  success.
- **robot prompt> ?-** Are you moving?
  yes.
- **robot prompt> ?-** Are you moving backwards?
  no.
- **robot prompt> ?-** Please stop.
  success.

Figure 3: A sample session in RCX-NLI.

DISCUSSION

Open source software has proven itself as a credible and powerful educational robotics solution. The popular open source programming alternatives for the LEGO RCX offer excellent packages for educational robotics at many levels. It has been argued that without open source software, the RCX would have never met the current levels of use it has reached. Similarly, our undergraduate projects would not have been so easily realized without the availability of open source code and easily available tools.

As educational robotics becomes more popular, other solutions will surface. For instance, with the emergence of the real-time Java specification, the idea of programming robots in Java will soon be a widespread reality. Systronix is developing the JCX, a Java-based replacement for the RCX. The JCX relies on a native Java byte-code processor made by aJile along with other hardware to offer a very powerful robotics controller. [36] Systronix, obviously observant of the strong community that has formed around the RCX, is being very open about their software and schematics.

While we have concentrated on the LEGO RCX, we must also address the increasing use and development of open source software for higher-end robotics. RT-Linux can be found in a few robotics projects, one of which is a walking humanoid robot. [19]

ActivMedia, the creator of the Pioneer series of mobile robots, has created a GPL library, ActivMedia Robotics' Interface for Applications (Aria), for controlling its robots. [15] Aria is dual-licensed also under a proprietary license, allowing researchers and companies to add proprietary features. There are also various online projects dedicated to creating open source robotics and control software. [29, 37]

The simple act of walking into a classroom with a robot can excite an otherwise listless room of students. Actually encouraging them to program the robot engages them in a way few pedagogical tools can. As more educators recognize the power of educational robotics, they must be encouraged to consider open source solutions to meet their needs. The best of the open source software is more robust than many commercial products, free, or of very low cost, very flexible, and encourages collaborative research and development.

REFERENCES

[1] Baum, D. Dave Buam's Definitive Guide to LEGO ® MindStorms™. Apress, 2000.

[2] Baum, D. Email interview. October 11 2001.

[3] Baum, D., Gasperi, M., Hempel, R., and Villa, L. Extreme MINDSTORMS™: An Advanced Guide to LEGO ® MINDSTORMS™. Apress, 2000.

[4] Beer, R., Chiel, H., and Drushel, R. Using Autonomous Robotics to Teach Science and Engineering. Communications of the ACM, June 1999.

[5] Fagin, F. Using Ada-Based Robotics to Teach Computer Science. In Proceedings of ITiCSE Annual Conference (2000).

[6] Hwang, D. J. and Blandford, D. K. A multidisciplinary team project for electrical engineering, computer engineering, and computer science majors. In Proceedings of the 2000 ASEE Annual Conference (2000).

[7] Knudsen, J. B. The Unofficial Guide to LEGO ® MindStorms™ Robots. O'Reilly & Associates, Inc., 1999.

[8] Kumar, D. and Meeden, L. A robot laboratory for teaching artificial intelligence. In Proceedings of the Twenty-ninth SIGCSE Technical Symposium on Computer Science Education (1998).

[9] Levesque H. J. and Pagnucco M. Legolog: Inexpensive Experiments in Cognitive Robotics. In Proceedings of the Second International Cognitive Robotics Workshop (August 21-22, 2000).

[10] Linder, S., Nestrick, B., and Mulders, S. Facilitating Active Learning With Inexpensive Mobile Robots. In Proceeding of the Sixth Annual CCSC Northeastern Conference (2001).

[11] Patternson-McNeill, H. and Binkerd, C. L. Resources for using LEGO ®

MindStorms™. In Proceedings of the Twelfth Annual CCSC South Central Conference (2001).

[12] Pennington, J. SPARK project. http://eggo.sourceforge.net/spark.php [accessed February 2002]

[13] Santalesa, R. "Installing TiVo." ZDNet Reviews. September 12, 2000. http://www.zdnet.com/products/stories/reviews/0,4161,2582375-2,00.html. [accessed February 2002]

[14] Wolz, U. Teaching Design and Project Management with Lego RCX Robots. In Proceedings of SIGCSE Annual Conference (2000).

[15] ActivMedia Robotics' Interface for Applications Application for ActivMedia Robots (ARIA). http://robots.activmedia.com/aria/ [accessed February 2002]

[16] Agenda Computing. http://www.agendacomputing.com/ [accessed February 2002]

[17] The Fast Light Toolkit. http://www.fltk.org [accessed March 2002]

[18] Festival Speech Synthesis System. http://www.cstr.ed.ac.uk/projects/festival/ [accessed March 2002]

[19] Perception-Action Integrated Humanoid Robot. University of Tokyo's Jouhou System Kougaku Laboratory (JSK Lab) and the Aircraft and Mechanical Systems Division of Kawada Industries, Inc. (Tokyo, Japan) http://www.jsk.t.utokyo.ac.jp/research/h6/H6_H7.html [accessed February 2002]

[20] Librcx. http://graphics.stanford.edu/~kekoa/rcx/tools.html [accessed February 2002]

[21] legOS. http://legOS.sourceforge.net/ [accessed February 2002]

[22] Lego/Scheme http://www.cs.indiana.edu/~mtwagner/legoscheme/ [accessed February 2002]

[23] leJOS. http://leJOS.sourceforge.net/ [accessed February 2002]

[24] Legolog http://www.cs.toronto.edu/cogrobo/Legolog/ [accessed February 2002]

[25] Lego::RCX:pm. http://members.home.com/quillan/lego/rcx.pm.html [accessed February 2002]

[26] MindControl http://homepages.svc.fcj.hvu.nl/brok/legomind/robo/ [accessed February 2002]

[27] Not Quite C. http://www.enteract.com/~dbaum/nqc/ [accessed February 2002]

[28] Not Quite C for Agenda Computing's VR3. http://brickbots.com/agenda/nqc.asp [accessed February 2002]

[29] Open Source Robot Control Software (OROCOS). http://www.orocos.org [accessed February 2002]

[30] Open Source Initiative. http://www.opensouce.org/ [accessed January 2002]

[31] Open Source Initiative. Frequently Asked Questions. http://www.opensouce.org/advocacy/faq.html [accessed January 2002]

[32] PbForth. http://www.hempeldesigngroup.com/lego/pbFORTH/ [accessed February 2002]

[33] Pylnp. http://www.hare.demon.co.uk/lego/pylnp.html [accessed February 2002]

[34] RCX Java API. http://www.escape.com/~dario/java/rcx/ [accessed February 2002]

[35] RCXPort. http://www.slewis.com/rcxport/ [accessed February 2002]

[36] Systronix's JCX. http://jcx.systronix.com/ [accessed February 2002]

[37] Teambots. http://www.teambots.org [accessed February 2002]

[38] TinyVM. http://tinyvm.sourceforge.net/ [accessed February 2002]