Impact of Libre Software Tools and Methods in the Robotics Field

Pablo Barrera, Gregorio Robles, José M. Cañas, Francisco Martín, Vicente Matellán
Grupo de Sistemas y Comunicaciones (GSyC)
Universidad Rey Juan Carlos
28933 Móstoles, Spain
{ barrera, grex, jmplaza, fmartin, vmo }@gsyc.escet.urjc.es

ABSTRACT

Software is one of the major components of robots; in fact, it is the main bottleneck for the proliferation of robotics in our everyday lives. In the last years the field of robotics has been an emerging application area of the libre (free/open source) software phenomenon. Libre software tools have been traditionally popular among the robotics research and teaching community. Even companies whose main business model is to sell robots have found convenient to share the software in order to promote a community around their products. In this paper we analyze the situation of libre software in these three subareas: industry, teaching and research. In particular, we describe commercial robots like the Cye and the Pioneer, a software platform like Orocos as a examples of industrial world applications, the libre tools around the LEGO Mindstorms in the case of teaching, and the Robocup competition and the Player/Stage platform in research area. All these cases show that libre software can act as a catalyst in the robotics industry which is still a sector in its early research and industrial stages.

Keywords

libre software, robotics industry, robotics research, robotics education, open source.

1. INTRODUCTION

Libre software[1] is mostly known due to successful projects such as the Apache web server, the Linux kernel, the OpenOffice.org office suite or any of the many libre programming languages such as Perl and Python. But beyond these applications, there is a vast activity in other fields which maybe are not mainstream but have been impacted by libre software and its development methods and tools. The robotics field is without doubt one of these application spaces.

The perception that robots are considered as tools or toys that can be modified or programed (i.e. like computers) by their users and that innovation can be performed on them has been a key factor in the emergence of communities around some robots. Many robotic kits sold include some kind of software to control or interact with the robots. This software can be adapted or improved to add new functionality that has not been included by the robot seller. Communities share improvements, solutions, code, and ideas with the help of the Internet. There even exist periodically organized tournaments where different groups compete on the abilities that they have implemented into their robots. Hence, in the last years several companies have focused their marketing strategies towards the creation and support of communities around their products. Besides technical feedback this enforces the fidelity to the branch name and increases its reputation. Users have been attracted to libre software development model both for ethical [13] and practical [12] reasons.

In this paper we will review the situation of libre software in the robotics field. We will touch three subareas of the robotics market: industry, education and research. Although they are treated independently, the reader will notice that there exist strong links between these three subareas. For each subarea we have selected some case studies known to the authors where libre software has been adopted. In the next section we will analyze the position of the robotics industry towards libre software. We have therefore selected three examples: the Cye robot from Probotics, the Pioneer robot from Activmedia and the Orocos project, an European Commission funded project. The third section is devoted to education and will be centered in the most widely used robots, the LEGO Mindstorms. In the fourth section we will have a look at the research initiatives, with emphasis in Robocup competition and software platform as Player/Stage. Finally, the last section presents conclusions and some predictions about the evolution of the libre software phenomenon in robotics that we envision.

2. LIBRE SOFTWARE IN THE ROBOTICS INDUSTRY

For a long time robotic applications have met the industrial world. Although this is still an evolving market a lot of en-

[1]Through this paper we will use the term "libre software" to refer to any code that conforms either to the definition of "free software" (according to the Free Software Foundation) or "open source software" (according to the Open Source Initiative).

terprises already center all their activity in robotics applications. Nowadays it is normal to find robots working inside factories, soldering, carrying, etc. But this is a limited market were users are only interested in final products, not in their modifications and adjustments, so we will not discuss this case.

This is not the only case inside the robotic market. Robotic enterprises have also spread to research, edutainment (education and entertainment), and service markets, so it is not difficult to find robots in places like stores, museums or even offices. Although companies working with expensive robots (like the ones used in factories) do not consider libre software as an alternative, new markets where the number of robots sold is higher offer heavy arguments for a company to enter in a model characterized by sharing source code that makes their robots work.

From our point of view, a software company, and by extension a robotics company, may obtain *better* software by doing this. This means for instance that software is more adjusted to the users' needs, bug detection and correction is faster, etc. On the other hand, users may obtain an open market, where more than one company can provide support for the same product. Users even can, if they will, get involved in the development, and in summary get customized products and better support.

In this section we will see some examples of business strategies around robots, focusing our attention in the Cye robot and the Pioneer robot. We will also talk about the Orocos software platform which aims to be a standard in the robotic industrial applications.

2.1 The Cye robot from Probotics

In the last years of the 90s some robotics companies started to realize that there were other ways of managing the software part of the robotic business. One example for this is given by the Cye robot[2] manufactured and distributed by Probotics Inc. At the beginning, this company sold its *Map-N-Zap* software independently from their robots. But soon they realized that the robots were not useful without that software, so they decided to include their software in the basic robot kit. In January 2000 they finally decided to distribute the software under terms of the GNU GPL (GNU General Public License). This suite included a GUI (Graphic User Interface), an iconic programming language, and the code for the robot communication protocol. The announcement of this strategic movement targeted clearly the libre software community[3]:

> "This decision was made in part because of the numerous requests from the Linux community for our source code, and the realization that they, and developers for other platforms, have much to offer to our mission, which is to make really cool robotic technology."

This sentence summarizes some of the key ideas behind the libre software business model that many robotic companies have adopted in the last years: the cost of porting their products to other platforms, or the maintenance, documentation and constant improvements needed in the software products they create is so high that is difficult for a company to afford it alone. In the libre software model some of these tasks can be delegated to a community of developers which has not to be in-house. In order to get such a community a good strategy is to guarantee the freedoms over the software that libre software provides.

Besides end-users, there is an important group which fosters heavy innovation in the robotics field and that can be attracted by such a libre software strategy: researchers. In the case of the Cye robots community, research groups from various universities, and specially from Carnegie Mellon[4] [1], are very active members. Probably these groups would have chosen a different robot if Probotics Inc. would not have released the software and the hardware specifications.

Giving the code for free would not have been enough for these research groups as for their purposes they need to modify the software and possibly redistribute their modifications. So, many of the behaviors that have been implemented for these robots, like dancing, racing or scaring pets would with high probability not be given for the Cye robot. Company and community work in a synergetic manner, focusing the former in improving the (hardware) product and letting the later get in charge of other tasks and providing feedback with new ideas and business opportunities.

2.2 The Pioneer robot from Activmedia

The Pioneer robot from Activmedia is sold as a research tool and is used in a lot of universities as the main physical research platform. By the time Activmedia started to sell this robot there were other alternatives in the market, which had similar characteristics and had been proven to be good enough for research purposes, but that had a restrictive software strategy. Nowadays Activmedia is the most known robot seller in the research market.

Probably the derminant difference was that Activmedia started to distribute its robot control software, Aria[5], as libre software. This software was a complete rewrite of their older Saphira development platform released under the GPL license. One of the goals of this project was offering legacy support for the earlier platform and the clients that were still using it. Aria has also been improved by contributions from users making the system work better.

Research groups that can not afford to buy a robot may start using high quality simulators and the Aria platform. for their developments which are completely libre. Later on when these groups have the possibility of buying a robot their natural choice is to select the Pioneer instead the other solutions. All software that has been developed for the simulators can be used in the real robot without or with minor modifications. Hence, this is a significant case where the business model of an enterprise is based on libre software.

2.3 The Orocos project

[2]http://personalrobots.com

[3]The announcement of the libre release can be found in http://slashdot.org/article.pl?sid=00/01/24/110219

[4]http://www.teambots.org/

[5]http://www.activrobots.com/SOFTWARE/aria.html

In modern environments where robots can be found from an industrial plant to an office, it is very important to have a common framework to develop robot applications, like those that are given for other software technologies as computer graphics or distributed computing.

The Orocos (Open RObot COntrol Software) project [3] presents a libre software framework for this kind of applications, offering generic functionality for machine tools and robots. It was born inside the EURON, the European Robotics Network mailing list and sponsored by the European Commission (EC) in 2000. When the EC sponsorship finished, the project was continued by its partners with multiple major software releases over the last two years.

The main goals of the project are to develop a feedback control software under a libre software license, being independent from any architecture, contribute to the development of programming interfaces for feedback control communities and to contribute to the development of free educational material.

Beside its robotics roots, the real-time Orocos framework has grown into the machine control field. Currently there are two main sub-projects inside the Orocos frame:

- **Open Real-time Control Services:** This is a general project, outside the robotic frame, developing real-time kernel for all possible feedback control applications.
- **Open Robot Control Software** This is an robot-specific project, with the aim of developing an application framework, offering motion generation and interpolation, kinematics and dynamics, control algorithms, estimation and identification, etc.

Currently the Orocos project continues with funds of the Flanders Mechatronics Technology Centre, developing the real-time framework. and coordinating the integration of Orocos in industrial machines of machinetool builders. This project can be considered a milestone in the robotics industry first for its impact in the industry, it was the first time that the EU funded libre software development in this industry; and second because it has been the starting point for others such as Orca[6], Ocean[7], etc.

3. LIBRE SOFTWARE IN ROBOTICS EDUCATION

Libre software has many benefits for its use in computer science education as previous literature has already reported [5]. Robotics courses are not an exception [8] [11].

Our research group has some experience with teaching the basics of robot control programming to computer science students. Apart from the syllabus, some other requirements have to be considered for these courses, as for instance the most appropriate robot and the most convenient tools that should be used to program the robots. In this section we will describe the tools that have been chosen for our teaching purposes, analyzing the reasons that were behind our decisions.

[6]http://orca-robotics.sourceforge.net

[7]http://www.fidia.it/english/research_ocean_fr.htm

3.1 The LEGO Mindstorms

LEGO Mindstorms[8] were originally conceived as an educational product for kids but right now it is without any doubt the most widely spread robot kit in any educational environment. The success of this product is partly because of the community that has been built around it and that has developed a completely new operating system (BrickOS [10]) and a programming language (NQC [2]), moving away from the closed tools provided by the manufacturer. The developments by the community have overcome the limits of the kits sold by LEGO raising Mindstorms to their privileged position.

Our group has chosen this platform for our university *Robótica*[9] (Robotics) course, that comprises two parts: first, building the robots; and second, programming then. The main reason was that it is a very flexible platform, being at the same time a complete end-user product at a very convenient price. Building a robot using the LEGO kit does not require any soldering, and the building blocks are intuitive and well known to all the students. The brain of the robot, the RCX, is based on a Hitachi H8/300 chip shipped with 32Kbytes of RAM, three output ports to connect motors, and three input ports where sensors can be connected.

The construction of a LEGO-based robot requires basic notions of mechanics, but small manuals as the official Constructopedia included in the kits, or the one authored by Fred Martin[10] have proved to be enough for our students.

There are different options for programming the LEGO Mindstorms robots [7]. On one hand, we have the software environment included in the Mindstorms 2.0 kit, the RCX code, which is a graphical programming language designed for kids. It is a very intuitive tool but a very limited development system, mainly used in introductory programming courses for children. In addition, it is not available as libre software and thus it is often not flexible enough for our intentions. Other options, such as NQC and BrickOS, are therefore considered.

3.2 The Not Quite C (NQC) language

NQC [2] stands for Not Quite C, and is a simple C-like syntax language that can be used to program LEGO's RCX programmable brick (from the Mindstorm set). It is the simplest text alternative to the drag and drop icon programming that Mindstorm provides.

NQC is libre software, released under the Mozilla Public License (MPL), but as it uses the standard operating system developed by LEGO it depends on a non-libre solution. However, the main drawback of NQC is that it has been designed to be simple and accessible, so that it is affordable for people with few programming knowledge. This means that there are strong limitations which include (but are not limited to):

- Subroutines do not admit parameters.
- Subroutines do not return values.

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

[9]http://gsyc.escet.urjc.es/docencia/asignaturas/robotica/ (in Spanish)

[10]http://constructopedia.media.mit.edu/

- There exist only global variables.
- The number of variables is strongly limited to just 32.
- There are no data structures neither static nor dynamic.

This made us discard NQC because our students are supposed to have strong programming abilities that they should use in this course to generate sophisticated behaviors for the robots.

3.3 The BrickOS operating system

The software we recommend to our students is the BrickOS operating system. BrickOS[11] is a libre software embedded operating system designed for the LEGO Mindstorms brick, an evolution of LegOS that was mainly designed by Markus Noga [10]. Compared to the standard software (the one from LEGO) BrickOS offers vastly superior performance and flexibility. The main features of the current version of BrickOS are the following:

- Dynamic loading of programs and modules.
- Full infrared (IR) packet networking.
- Preemptive multitasking.
- Dynamic memory management.
- Drivers for all RCX subsystems.
- 16 MHz native mode speed.
- Access to the RAM.
- Full use of programming languages, for instance C (pointers, data structures, etc.).

The development environment in our labs is made up of a Debian GNU/Linux machine which compiles and communicates with the Mindstorm robot, both to download and to debug programs. BrickOS works reliably with several GNU/Linux distributions. Using it requires the installation of a cross compilation system, based on `binutils` and `gcc`.

BrickOS is just the operating system, so obviously programming languages are needed. In the course we use C, but any other language whose compiler supports cross compilation can be used; for instance, most people use C++ instead of C.

In that environment, that is Mindstorm brick and BrickOS, it is sometimes difficult to debug a program, so we have installed a couple of libre simulators for our students: *Emulegos*[12] and *LegoSim*[13]. Both provide a graphical interface for the debugging process but have different features. With *LegoSim* you can communicate to a real robot and monitor its behavior while EmuLegOS provides *real world* emulation, a place to put code to mimic some of the mechanical features of the robot, such as a rotation sensor that turns while the motor is running. There exists also the possibility of controlling the status of the virtual ports connected to the robot.

[11]http://brickos.sourceforge.net
[12]http://emulegos.sourceforge.net
[13]http://moss.csc.ncsu.edu/∼muller/legosim

4. LIBRE SOFTWARE IN ROBOTICS RESEARCH

In this section we will briefly review the current state of libre software tools and methods in the robotics research world. Libre software offers scientists the possibility of sharing with their peers how they have solved problems. This, of course, is basic for the evolution of science; but libre software goes beyond that and makes this happen in a cost-effective way [9].

In this section we will talk about how libre software can improve research results in a challenging environment like the Robocup competition. We will also see how libre software programming environments emerge to make the task of programming in heterogeneous hardware using abstract platforms easier. There are several software platforms and many of them are libre software as for instance Miro[14], Marie[15], Carmen[16], JDE[17], and Player/Stage. We will focus our attention in the last one which has become the reference platform in the research world.

4.1 The Robocup competition

As an example of libre software practices in the robotics research we can take a look at the Robocup competition [6]. Robocup[18] is an international research and education initiative. Its goal is to foster artificial intelligence and robotics research by providing a challenging problem (playing soccer as a primary domain) where a wide range of technologies can be examined and integrated.

The organization of the Robocup consciously fosters the use of libre software as a way of improving the level of the competition. All software produced by the organization is therefore released under a libre software license and most of the teams do share their code, mainly under a GPL or LGPL license.

Sharing code has allowed teams of recent creation to participate from the beginning at a similar level than the rest of the teams and to focus their efforts on the improvement of relevant state-of-the-art techniques. A positive aspect of this policy is the increasing advance of research since no time is wasted in already solved problems. But the advantages of libre software are not limited only to newcomers; competition has made that teams have improved their software reaching results hardly believed five years before. The organization promotes this process by establishing more restrictive rules each year.

4.2 The Player/Stage platform

An important component of the development with robots is the software architecture. There are several robots from different manufacturers that have completely different hardware and their own development environment. To ease this task software platforms which abstract from the lower details have appeared. They allow to program several robots using a common API. The development of these kind of tools is not the main goal for research, however good tools can improve

[14]http://smart.informatik.uni-ulm.de/MIRO/content.html
[15]http://marie.sourceforge.net/
[16]http://www-2.cs.cmu.edu/∼carmen/
[17]http://gsyc.escet.urjc.es/∼jmplaza/software.html
[18]http://www.robocup.org

both quality and cost. Libre software has also some advantages here. By sharing their tools different groups tackle only small changes on the complete system, gaining from the work done by the rest of the community. As an example we can take the Player/Stage project [4].

Player/Stage[19] began at the USC Robotics Research Lab in an investigation about multiple robot systems. They needed a tool for programing, interfacing, and even simulating these kind of systems. The project right now provides tools that simplify controller development for multiple or distributed robots. Since the beginning, the development has been open to every contribution providing their software under the GPL license. Original authors knew that users of their software were also developers that could contribute with their work to get a better product. Due to this the project was used by more researchers around the world, building a great community. Player/Stage works perfectly in GNU/Linux and supports tens of commercial robot models. Contributed libraries also have bindings for languages such as C, C++, Python, C# or Java. The support for multiple robots and programming languages has converted Player/Stage in the *de facto* standard platform for research development.

The project is divided in two parts:

- Player provides a software platform to develop control application independently of the robot hardware. This allows researchers to focus on their programs avoiding the hardware dependent tasks or portings.
- Stage can be used as a complement for Player. It offers a simulated 2D world where a high number of robot simulated by Player can interact.

As the evolution of the project continues new components have been added. This is the case of Gazebo, which is similar to Stage but offers a complete 3D world simulation. Gazebo is also able to give simulated images for controllers that use visual information.

The diffusion of Player/Stage has increased in a way that it has become the reference platform for many research teams. Hence, new tools arise as visualizer for Player or control architectures built over the complete system.

5. CONCLUSIONS

Robotics is an emerging sector of the computer industry that is in its early research and application stages. In this paper we have presented some examples of why libre software can act as a catalyst for the development of this industry.

There are many benefits that can be derived from the use of libre software development models and structures in the robotics field and both users and enterprises are starting to realize about it. In fact, some enterprises have built their business model around the use of libre software as we have seen in this paper.

We have shown examples where the use of libre software has been the key factor to make a product the reference of a complete market, both in research and education. In both cases libre software is easier to adapt to the particular needs of users and to reuse it in other situations. The ability to adapt software is used also to extend the potential users base and to fulfill their needs. Libre software communities have even brought tools that have been shown to be more popular than the ones provided by the original manufacturers.

Finally we have presented an example where the use of libre software has allowed to achieve faster developments and better results. By sharing the already solved solutions, new developers can center on improving the parts of the system which are not that mature.

[19]http://playerstage.sourceforge.net

6. ACKNOWLEDGEMENTS

The work of Pablo Barrera has been supported by Universidad Rey Juan Carlos. The work of José M. Cañas, Francisco Martín and Vicente Matellán has been partially supported by the Spanish MEC project ACRACE, (DPI2004-07993-C03-01). The work of Gregorio Robles has been funded in part by the European Commission, under the CALIBRE CA, IST program, contract number 004337, and in part by the Universidad Rey Juan Carlos under project PPR-2004-42.

7. REFERENCES

[1] **Parag Batavia and Illah Nourbakhsh** *Path Planning for the Cye Robot.* Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2000, October 2000, Vol. 1, pages 15-20.

[2] **Dave Baum**. *Dave Baum's Definitive Guide to LEGO Mindstorms.* Apress, USA, 1999.

[3] **Herman Bruyninckx** *Open robot control software: the OROCOS project.* Proceedings of the International Conference on Robotics and Automation, 2001. Vol 3. Pages 2523- 2528.

[4] **Brian Gerkey, Richard T. Vaughan and Andrew Howard** *The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems* Proceedings of the 11th International Conference on Advanced Robotics, pages 317-323, Coimbra, Portugal, June 2003 (ICAR'03).

[5] **Jesús M. González-Barahona, Pedro de-las-Heras-Quirós, José Centeno-González, Vicente Matellán-Olivera, and Francisco J. Ballesteros**. *Libre software in CS practice teaching (The experience at Carlos III University).* IEEE Software, Vol. 17, No. 3, pp. 76-80, May/June 2000.

[6] **Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda and Eiichi Osawa** *RoboCup: The Robot World Cup Initiative.* Proceedings of the first international conference on Autonomous agents table of contents, 1997. Marina del Rey, California, United States. Pages 340-347.

[7] **Jonathan B. Knudsen**. *The Unofficial Guide to LEGO MINDSTORMS[tm] Robots.* O'Reilly, 1st edition edition, 1999.

[8] **Deepak Kumar and Lisa Meeden**. *A Robot Laboratory for Teaching Artificial Intelligence* Proceedings of the Twenty-ninth SIGCSE Technical Symposium on Computer Science Education (SIGCSE-98), February 1998. Pages 341-344.

[9] **Tony Meyer**. *Building Cost-effective Research Platforms: Utilising Free — Open-source Software in Research Projects*. Res. Lett. Inf. Math. Sci. (2003) 4, 91-99. `http://iims.massey.ac.nz/research/letters/volume4/10meyer.pdf`.

[10] **Markus L. Noga**. *Open-source embedded operating system for the LEGO Mindstorms.* `http://www.noga.de/legOS/`.

[11] **Keith O'Hara and Jennifer S.Kay**. *Investigating Open Source Software and Educational Robotics*. The Journal of Computing Sciences in Colleges, Volume 18 , Issue 3 (February 2003), Pages 8-16. `http://elvis.rowan.edu/∼kay/papers/OSSEduRob.pdf`.

[12] **Eric S. Raymond**. *The Cathedral and the Bazaar*, 1998, Available at `http://catb.org/∼esr/writings/cathedral-bazaar/`.

[13] **Richard M. Stallman**. *Why Software Should Not Have Owners*. 1998. Available at `http://www.gnu.org/philosophy/why-free.html`.