Libre[*] software role in edutainment robotics market

Vicente Matellán Olivera, Jesús María González Barahona,
José Centeno González, and Pedro de las Heras Quirós
E-mail: {vmo,jgb,jcenteno,pheras}@gsyc.escet.urjc.es
Grupo de Sistemas y Comunicaciones (GSyC). Fax: +34 91 664 74 90
Dept. Ciencias Experimentales e Ingeniería. Universidad Rey Juan Carlos
C/ Tulipán s/n 28933-Móstoles (Madrid) - SPAIN

Abstract

Some of the most successful products in the market of robots for edutainment have made manufacturers realize that software for controlling the robots is one of the key elements of the success. Some of them have also conclude that *libre* software is the best way to produce, maintain, and distribute their software. This paper will summarize the fundamentals of the *libre* software model, and their benefits for the edutainment robot market, as well as some examples of companies currently supporting it.

Keywords: *libre* software, open source, free software, business model.

1 Introduction

Which one is the key factor for succeeding in the edutainment robots market? This is of course the key questions in any market, and the obvious answer is 'a good product', and 'better marketing'. Our point, and the main idea of this paper, is that the software involved is one of the key elements in the definition of a successful edutainment product, and that the way in which software is licensed is one of the critical issues both in the quality, and in the marketing strategy. Moreover, we think that open source model is the best way to produce and to distribute software in general, and software for robots in particular.

We will try to explain in this paper how the open source development model can produce better software, and how it do it in a more efficient way. This has been probed in the traditional software market, where products as Apache, GNU/Linux, StarOffice, etc. have shown that it is possible to produce competitive products using open source models. We will analyze if this can be also achieved in the edutainment robotics field.

This paper will be focused in the segment of the small robotic toys. We think that the spread of robots in our lives is going to happen as robotic toys instead of as service robots, or production robots, as once was foreseen. This does not mean that the number of industrial robots is not growing [9], this mean just that they have evolved more slowly than expected. We also think that the *libre* software can also be useful for service robots.

These robotic toys are not used just for leisure. For instance, they can be used as educational resources. So, dozens of universities are using robotic toys as the LEGO Mindstorms both for teaching or researching purposes. Of course, these robots are mainly sold just for leisure. Other robots that are considered part of the edutainment market, as the the ones used in entertainment facilities (let's say thematic parks, or museums) will never become more than a small fraction of the market. Of course, *libre* software is also applicable to these robots.

In summary, in this paper we will try to show that the development of software for small robotic toys is suitable for the open source approach. To do that, the paper will be organized as follows. In the first section we will present some successful stories of traditional software products that show how *libre* software can be used. In the second one the implications of this model of producing and distributing software in the robotics field are analyzed. The last one summarizes the previous ideas, and tries to foresee the evolution of the *libre* software in edutainment robotics.

[*] *Libre* word will be used in this paper to avoid the usual misunderstanding between "free" as in "free beer" (*gratis* in Spanish), and *free* as in "free speech" (*libre* in Spanish)

2 *Libre* software

In the first years of the digital computer history, that's it during the 60's, the software was freely shared among computer programmers. It was also improved by means of collaboration among the few companies that share the same computer. Obviously the manufacturer gave the source code of its programs to its clients to make it easier the improvement of the programs, and in the same way its competitive advantage. This changed after the unbundling, which became software in a new "product" that can be sold separately from the computer.

During the seventies and the eighties a strong business based on selling copies emerged. Companies as Microsoft, Oracle, Corel, etc. became huge multi-national corporations. However, for some people it became clear that there are some other ways of getting money from software (apart from the selling of copies), and also that this other ways will produce better software, both in price. The most widely know claim for a different way of creating software was the "The GNU Manifest" [8]. Basically, Richard Stallman contribution was the idea of *copyleft*: using the copyright legislation to ensure that the source code of programs was available, and freely distributable, and modifiable.

For some time this way of producing and distributing software was considered anti-economic, some kind of altruism. However, the last year have shown that this is not only a different way of producing software, but also of making money. Companies as RedHat, VA-Linux, or Andover have got successful IPOs in NASDAQ, and most of them are making money.

The first sign in this change was the release as open sources of the Netscape Navigator in 1998. In that moment Frank Hecker[3] wrote his paper "Setting up the shop: The Business of Open-Source Software" where he identified some ways of making money into the *libre* software model.

Support sellers: Revenue comes from media distribution, branding, training, consulting, custom development, and post-sales support.

Loss leader: Loss leader of traditional software (as Netscape vs. Microsoft in the web browsers market) would turn into open source because:

- *Libre* software helps building vendor brand and reputation.
- Makes traditional products more functional.
- Increases the overall base of developers and users familiar with vendor's product line.

Widget sellers: Companies that are in business primarly to sell hardware. They will use libre software for drivers or interfaces. VA-Linux is a well-known example of a company selling PCs running GNU/Linux.

Accessorizing companies: companies which distribute books and other physical items associated with and supportive of libre software. For instance, O'Reilly & Associates is selling books about LegOS, the open source operating system for the LEGO Mindstorms, or about NQC, a reduced version of C programming language for the same robot.

Brand licensing and software franchising, while retaining the rights to the libre product trademark. Can charge other companies for using the trademark in creating derivative products.

There are some other ways that are currently being explored by companies as Sun, which has recently free StarOffice (using the GPL license), or web-based ones as SourceForge or CollabNet that will probably evolved into succesful business initiatives.

Some of these ways of making profits are really well suited for the robotics field. For instance, the widget sellers model proposed by Hecker is exactly what robotics companies do. Most of the robotics companies get their money from selling the robots, not the software. Besides, the ones which sell the software will probably sell more robots if the software was free (both *libre* and *gratis*). This is specially true in robots for entertainment, and also in robots for research or educational purposes, and it consequences will be analyzed in the next section.

3 *Libre* software and edutainment robotics

In this moment, edutainment robots are still considered as toys for *geeks*, and this kind of people is used to get involved in the development of their toys. They consider themselves as part of "communities", and a successful marketing strategy is to create one of these communities around the company products. The use of *libre* software will enforce this feeling, increasing the reputation of the company, and consequently the fidelity to the branch name. However, this is not the main advantage of distributing freely the source code. From our point of view, a software company, and by extension a edutainment robotics company, may obtain **better** software. 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 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.

However, the vast majority of the robotic companies do not consider this alternative. They still think that information products (like software) are equivalent to material products (let's say bread), and try to obtain their benefit from "selling copies" or "new releases" of their product. We think this is not the right way, both by ethical[7], and economical[6] reasons. The libre (open source) software model model is being recognized as a real alternative for the production and distribution of software. Apache is the most widely used web server, GNU/Linux is the second and the fastest growing operating system, Perl or Python programming languages are becoming some of the more popular ones, ...

In this direction, some companies have started to realize that there are other ways of managing the software part of the robotic business. Let's consider the example of Cye [1] robot that could be considered a canonical evolution towards *libre* software in a robotics manufacturer. Probotics Inc., the company that manufactures and distributes it started selling its *Map-N-Zap* software, but soon they realized that the robot was not useful without that software, so they decided to include the software in the basic robot kit. In January 2000 they decided that the software will be distributed under GPL (GNU Public License). This included all the GUI (Graphic User Interface), the iconic programming language, and the robot communication protocol code. They stated that[2] :

> "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 model: no company can afford the costs of porting their products to every platform, of customizing it to every user preferences, or the maintenance, documentation, and constant improvements needed in software products. In the *libre* software model some of these tasks can be delegated in the community of developers. In order to get a community of developers involved, some freedoms have to be guaranteed. These freedoms make up the fundamentals of the *libre* software model and can be summarized in three:

1. Freedom to use the software in any way: in every country, in any computer, etc.

2. Freedom to modify the software in order to adapt it, improve it, etc. which means that the source code have to be provided.

3. Freedom to re-distribute the software in any way, for free or not.

We have spoken mostly about users when describing the community around a *libre* software product, but it also include researchers. For instance Carnegie Mellon [3] robotics groups are active members of the Cye developers community. Probably, these groups would have chosen a different robot if the company wouldn't have released the software and hardware specifications, or if they would have done it in a way that do not fulfill the previous freedoms.

Would have Probotics Inc. developed programs for dancing, racing, or scaring pets [4]? We don't think so, but some of the users have done it and they are sharing them with other users. That's one of the nice parts of having a community around a product. The company can focus on improving the product, while the users can get in charge of other tasks. In a different way, Carnegie Mellon researchers can contribute to the development of the product by providing new ideas, as M.I.T. did in the case of the LEGO based robots.

Let's consider that case, the LEGO Mindstorms kit, without any doubt the most widely spread robot kit until now. Why has this product been so successful? From our point of view, it is because of their ability to build a community of developers around it. This community have developed a completely new operating system (legOS[4]) and a programming language such as NQC [1], which stands for Not Quite C (a simple language with a C-like syntax that can be used to program LEGO's RCX programmable brick, from the Mindstorms set).

[1] http://personalrobots.com

[2] The whole letter to the community can be read in http://personalrobots.com/developers/opensource/

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

[4] Some of these behaviors are described in http://tds.simplenet.com/cye/

Will the Sony AIBO[5] become such a phenomenon as the Mindstorms? This robot got a huge impact on the media. However first versions were a completely closed product where the owners just can see the different performances of the electronic pet. New versions include software to "train" the robot, but this software, as well as the operating system, are distributed as proprietary software. There are no other options available, as far as we know, to improve or directly program the robot. This means that this robot is going to evolve much more slowly than others that take the *libre* way. For instance, let's think about the vast number of different sensors that have been designed for LEGO Mindstorms robots, the huge number of programs and applications, and try to think what will happen when the LEGO hardware is improved. We have to realize that AIBO is without doubt a technologically superior product, but from our point of view may be another fiasco, as it was the BETA video-recorder standard because a wrong strategy.

4 Conclusions

In summary, we think that *libre* software model for building and maintaining edutainment robots software is the best alternative, both from the companies and from the users point of view. We also think that the market will force companies to move into this model, as it is doing right now in the traditional software market.

Obviously, some work has to be done to adapt the *libre* software business models to the robotics environment. But, the major work is to explain companies as Sony or LEGO, that the right way to get more and better software, which obviously would become their products more likely for more people, is to use

References

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

[2] **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)*.

[3] **Frank Hecker**. *Setting Up Shop: The Business of Open-Source Software*. `http://http://www.openresources.com/documents/setting-up-shop`

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

[5] **Working group on Libre Software**. *Free Software / Open Source: Information Society Opportunities for Europe?*. `http://eu.conecta.it`. 2000

[6] **Eric S. Raymond**. *The Cathedral and the Bazaar*, 1998, Available in `http://tuxedo.org/ esr/writings/cathedral-bazaar/`

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

[8] **Richard M. Stallman**. *The GNU Manifest.*

[9] **United Nations Economic Commission for Europe** *The Boom in Robot Investment Continues - 900,000 Industrial Robots by 2003*. Press Release ECE/STAT/00/10. Geneva, 17 October 2000. `http://www.unece.org/press/00stat10e.htm`

[5] http://www.sony.com/aibo