

Modelos y herramientas de software libre en el campo de la robótica

Pablo Barrera, Gregorio Robles, José M. Cañas, Francisco Martín, and Vicente Matellán

Grupo de Sistemas y Comunicaciones (GSyC), Universidad Rey Juan Carlos,
C/ Tulipán s/n CP. 28933 Móstoles (Madrid), Spain.
{barrera,grex,jmplaza,fmartin,vmo}@gsyc.escet.urjc.es

Abstract. El software es uno de los principales componentes de los robots. Realmente es el principal cuello de botella para la proliferación de la robótica en nuestra vida diaria. En los últimos años el campo de la robótica ha sido un área en el que el fenómeno del software libre ha crecido de manera extraordinaria. En este artículo analizamos la situación del software libre en tres subáreas: industria, enseñanza e investigación. Las herramientas libres han sido muy populares entre la comunidad docente e investigadora en el campo de la robótica. Incluso compañías cuyo principal modelo de negocio es la venta de robots, han creído conveniente compartir el software con el fin de promover comunidades alrededor de sus productos. Todos estos casos muestran que el software libre puede actuar como un catalizador en la industria robótica, un sector aún en fase temprana de desarrollo, en el mundo investigador e industrial.

1 Introducción

Conocemos principalmente el software libre gracias a proyectos muy difundidos como son el servidor Apache, el kernel de Linux o la *suite* ofimática OpenOffice.org, así como múltiples lenguajes libres, entre los que cabe citar Perl o Python. Pero más allá de esas aplicaciones, existe una amplia actividad en otros campos, quizá menos conocidos, pero también fuertemente influenciados por el software libre, sus modelos de desarrollo y herramientas. La robótica es claramente uno de estos campos.

Muchos de los kits robóticos que se venden actualmente incluyen algún tipo de software para controlar o interactuar con el robot. Estos programas puede adaptarse, mejorarse o incluso reemplazarse para añadir nuevas funciones que no fueron incluidas por el fabricante. Las comunidades que se forman alrededor de estos robots no comparten solo el interés por ellos, sino también mejoras, soluciones, código y nuevas ideas gracias a Internet. Los usuarios se han sentido atraídos por el modelo de desarrollo del software libre tanto por razones éticas [13] como prácticas [12].

En este artículo revisaremos la situación actual del software libre en el campo de la robótica. Nuestro estudio ha dividido en tres partes el campo de la robótica: industria, educación e investigación. Para cada una hemos elegido algunos casos de estudio conocidos por los autores, donde el software libre ha sido la estrategia adoptada.

El resto del artículo está organizado de la siguiente manera: en la siguiente sección analizaremos el estado de la robótica en la industria a través de tres ejemplos: el robot Cyb de Probotics, el robot Pioneer de Activmedia y el proyecto Orocós, financiado por la Comisión Europea. La tercera sección se centrará en la educación, donde encontramos el que es sin duda el kit robótico más extendido: el LEGO Mindstorms. La cuarta sección tratará sobre las iniciativas de investigación, mencionando la competición Robocup y las plataformas Player/Stage y JDE. Finalmente, se presentan una serie de conclusiones y algunas predicciones acerca de cuál creemos que será la evolución del fenómeno del software libre en el campo de la robótica.

2 Software libre en la industria robótica

Durante mucho tiempo se han empleado robots en las aplicaciones industriales. Hoy en día no es raro encontrar robots en fábricas, soldando, transportando materiales, o en cualquier otra tarea repetitiva. En este mercado es difícil la entrada de nuevas ideas en él, y la cantidad de usuarios interesados en algo más que productos cerrados sigue siendo limitada.

A pesar de que las compañías que trabajan con robots caros todavía no han encontrado los alicientes suficientes para entrar a una dinámica dirigida por el uso de software libre, los nuevos mercados donde el número de robots

vendidos es muy alto, se ha demostrado claramente que existen muchos buenos argumentos para impulsar a una compañía de robots a entrar en un modelo caracterizado por compartir el código fuente que hace funcionar a sus robots.

En esta sección veremos algunos ejemplos de estrategias de negocio sobre robots, poniendo atención en los robots Cyé y Pioneer. También hablaremos de la plataforma software Orocos, que pretende ser un estándar en la aplicaciones de la industria de la robótica.

2.1 El robot Cyé de Probotics

En los últimos años de la década de los 90, algunas compañías de robots comenzaron a darse cuenta que había otras formas de gestionar su software. Un ejemplo es el robot Cyé¹ desarrollado y distribuido por Probotics Inc. Inicialmente esta compañía vendía su software *Map-N-Zap* independientemente de sus robots. Pero pronto se dieron cuenta que los robots no era útiles sin ese software, así que decidieron incluirlo en el kit básico del robot. En enero del año 2000 finalmente decidieron distribuir el software bajo licencia GNU GPL (GNU General Public License). El anuncio de esta estrategia tenía como objetivo claro la comunidad de software libre²:

“Esta decisión fue tomada en parte por la numerosa cantidad de peticiones desde la comunidad Linux de nuestro código fuente, y el hecho de que ellos, y desarrolladores de otras plataformas, tienen mucho que ofrecer a nuestra misión, que es hacer verdaderamente atractiva la tecnología de los robots.”

2.2 El robot Pioneer robot de Activmedia

El robot Pioneer de Activmedia se vende como una herramienta de investigación, y se usa por una gran cantidad de universidades como principal plataforma de investigación en robots móviles. En el momento en que Activmedia comenzó a vender este robot existían otras alternativas en el mercado, las cuales tenían características similares y se había comprobado que eran lo suficientemente buenas para fines investigadores. Actualmente Activmedia es conocido como el mayor vendedor de robots para fines investigadores.

Probablemente la diferencia determinante para esta evolución fue que Activmedia comenzó a distribuir su software de manera abierta mientras que las otras empresas eran más restrictivas. El software de control del robot, Aria³, era una nueva versión de su anterior plataforma de desarrollo Saphira, esta vez publicado bajo la licencia GPL. Una de las metas de este proyecto era ofrecer soporte a la plataforma anterior y a los clientes que estaban aún usándola. Aria ha sido también mejorada por contribuciones de usuarios, repercutiendo en la calidad y funcionamiento del sistema.

2.3 El proyecto Orocos

En la situación actual, en la que se pueden encontrar robots desde en plantas industriales hasta en oficinas, es importante tener un entorno común de desarrollo de aplicaciones para robots. Dichos entornos están presentes en otras tecnologías de software como el tratamiento de imágenes o la computación distribuida.

El proyecto Orocos [3] (Open ROBot COntrol Software) presenta un entorno de desarrollo que ofrece funcionalidad genérica para los robots y sus aplicaciones. Este proyecto nació dentro de EURON (European Robotics Network) y fue patrocinado por la Comisión Europea (EC) en el año 2000. Actualmente el proyecto Orocos continúa con fondos del Centro de Tecnología Flanders Mechatronics, desarrollando el entorno de tiempo real y coordinando la integración de Orocos en máquinas industriales.

El objetivo principal del proyecto es desarrollar una plataforma software de control bajo un licencia libre, manteniendo la independencia sobre arquitecturas concretas, contribuir al desarrollo de interfaces de programación de arquitecturas de control y al desarrollo de material docente libre.

Además de sus raíces en el campo de la robótica, el entorno de desarrollo de aplicaciones en tiempo real de Orocos se ha desarrollado en el campo del control de maquinaria. Actualmente hay dos subproyectos principales dentro del marco de Orocos:

¹ <http://personalrobots.com>

² el anuncio de la versión libre se puede encontrar en <http://slashdot.org/article.pl?sid=00/01/24/110219>

³ <http://www.activrobots.com/SOFTWARE/aria.html>

- **Servicios Abiertos de Control en Tiempo Real:** Este es un proyecto general, fuera del marco de la robótica, que desarrolla un kernel de tiempo real para todos las posibles arquitecturas de control.
- **Software Abierto de Control de Robots:** Este es un proyecto específico para robots, con la intención de desarrollar un entorno de desarrollo de aplicaciones, ofreciendo generación de movimientos e interpolación, cinemática y dinámica, algoritmos de control, estimación e identificación, etc.

Este proyecto puede ser considerado un hito en la industria robótica por su impacto en este campo: fue la primera vez que la Unión Europea aportaba fondos a un desarrollo de software libre en la industria; y segundo, porque ha sido el punto de partida para otros proyectos como Orca⁴, Ocean⁵, etc.

3 Software libre en la enseñanza de la robotica

El uso del software libre es beneficioso para la enseñanza de ingenierías, como se ha reflejado ya [5]. Los cursos de robótica no son una excepción [8] [11].

Nuestro grupo de investigación tiene experiencia en la enseñanza de los principios de la programación de control de robots a estudiantes de ingenierías informáticas. Además de buscar contenidos adecuados, es necesario elegir de manera adecuada el robot y las herramientas que deben usarse para su programación. En esta sección describiremos las herramientas que usamos para nuestros fines docentes, analizando las razones que hay detrás de nuestras decisiones.

3.1 LEGO Mindstorms

Los LEGO Mindstorms⁶ fueron originalmente concebidos como un producto para que los niños aprendieran, pero hoy en día es sin duda el kit de robots más distribuido en cualquier entorno docente.

El éxito de este producto es, en parte, por la comunidad que se ha formado alrededor de él y que ha desarrollado por completo un nuevo sistema operativo (BrickOS [10]) y un lenguaje de programación (NQC [2]), desplazando las herramientas cerradas proporcionadas por el fabricante. Los desarrollos de esta comunidad han sobrepasado los límites de los kits que vende LEGO, llevando Mindstorms a su posición privilegiada en el mercado.

3.2 El lenguaje Not Quite C (NQC)

NQC [2] son las siglas de Not Quite C, y es un lenguaje parecido a C que se puede usar para programar el ladrillo RCX de LEGO (del kit de Mindstorm). Es la alternativa textual sencilla al entorno visual que aporta Mindstorm.

NQC es software libre, publicado bajo la *Mozilla Public License (MPL)* pero, como usa el sistema operativo estándar desarrollado por LEGO, depende de un solución no libre.

3.3 El sistema operativo BrickOS

BrickOS⁷ es un sistema operativo embebido libre diseñado para el ladrillo de LEGO Mindstorms. Es una evolución de LegOS que fue principalmente diseñado por Markus Noga [10]. Comparado con el software estándar (el de LEGO), BrickOS ofrece extraordinarias mejoras de rendimiento y flexibilidad.

4 Software libre en la investigación en robótica

El software libre ofrece a la comunidad científica la posibilidad de compartir entre ellos las soluciones a diferentes problemas comunes. Esto, por supuesto, es básico para la evolución de la ciencia; pero el software libre va más allá y hace que el proceso suceda de una manera asequible [9].

En esta sección hablaremos de cómo el software libre puede mejorar resultados de investigación en entornos competitivos, como es la Robocup. También veremos cómo los entornos de programación de software libre se usan para

⁴ <http://orca-robotics.sourceforge.net>

⁵ http://www.fidia.it/english/research_ocean_fr.htm

⁶ <http://www.legomindstorms.com>

⁷ <http://brickos.sourceforge.net>

realizar tareas de programación de hardware heterogéneo usando plataformas abstractas de forma sencilla. Hay varias plataformas software y muchas de ellas son software libre, como por ejemplo Miro⁸, Marie⁹, Carmen¹⁰, Player/Stage¹¹ y JDE¹². Enfocaremos nuestra atención en las dos últimas. La primera se ha convertido en la plataforma de referencia en el mundo de la investigación y JDE es la plataforma desarrollada en nuestro grupo

4.1 La competición de la Robocup

Como ejemplo de prácticas de software libre en la investigación de la robótica, podemos echar una ojeada a la competición de la Robocup [6]. Robocup¹³ es una iniciativa internacional de investigación y docencia. Su meta es fomentar la investigación en el campo de la inteligencia artificial y de la robótica, proporcionando un problema desafiante (principalmente jugar al fútbol), donde una amplia gama de tecnologías pueden ser examinadas e integradas.

La organización del Robocup fomenta el uso del software del libre como manera de mejorar el nivel de la competición. Todo el software producido por la organización se libera bajo licencia de software libre, y la mayoría de los equipos comparten su código, principalmente bajo licencia GPL o LGPL.

4.2 La plataforma Player/Stage

Un componente importante del desarrollo con robots es la arquitectura software. Existen múltiples robots de diversos fabricantes que tienen hardware totalmente distinto y sus propios entornos del desarrollo. Con el fin de facilitar esta tarea han aparecido plataformas software que abstraen los detalles de más bajo nivel, proporcionando un API común para programar diferentes robots. El desarrollo de esta clase de herramientas no es la meta principal de la investigación, no obstante pueden mejorar la calidad y el coste de los resultados. El software libre presenta también algunas ventajas aquí. Compartiendo sus herramientas, diferentes grupos introducen cambios pequeños en el sistema completo, beneficiándose del trabajo hecho por el resto de la comunidad. Como un ejemplo tenemos el proyecto Player/Stage [4].

Desde el principio el desarrollo ha estado abierto a cualquier contribución, ya que proporcionaban su software bajo licencia GPL. Los autores originales sabían que los usuarios de su software eran también desarrolladores que podrían contribuir con su trabajo para conseguir un producto mejor. Debido a esto, el proyecto fue utilizado por más investigadores por todo el mundo, construyendo una gran comunidad. El soporte para diferentes modelos de robots y múltiples lenguajes de programación ha convertido a Player/Stage en la plataforma estándar *de facto* para desarrollos de investigación.

4.3 JDE

Sobre las plataformas citadas anteriormente (Aria y Player/Stage), se construyen arquitecturas de más alto nivel como JDE, que ha sido desarrollado íntegramente por nosotros.

JDE es una aproximación jerárquica a una arquitectura robótica de control, inspirada en la etología y basada en la activación selectiva de esquemas.

La implementación actual de JDE consiste en una infraestructura de software y una colección de esquemas específicos de comportamiento, que contienen técnicas particulares de control o de percepción. La infraestructura incluye los esqueletos de esquemas, los esquemas del servicio según el robots soportado, los mecanismos del software para la selección distribuida de la acción y la activación selectiva de esquemas, así como su ajuste. Además, algunas bibliotecas también se han escrito para manipulación de rejillas y control borroso, con el fin de hacer más fácil la generación del comportamiento. Todo el código se ha desarrollado en el lenguaje C para máquinas GNU/Linux, y está liberado bajo licencia GPL.

⁸ <http://smart.informatik.uni-ulm.de/MIRO/content.html>

⁹ <http://marie.sourceforge.net/>

¹⁰ <http://www-2.cs.cmu.edu/~carmen/>

¹¹ <http://playerstage.sourceforge.net>

¹² <http://gsync.escet.urjc.es/~jmplaza/software.html>

¹³ <http://www.robocup.org>

5 Conclusiones

La robótica es un sector emergente de la industria de las computadoras que todavía se encuentra en etapas tempranas de investigación y uso. En este trabajo hemos presentado algunos ejemplos de porqué el software libre puede actuar como catalizador para el desarrollo de esta industria.

Hay muchas ventajas que se pueden derivar del uso de modelos y estructuras de desarrollo de software libre en el campo de la robótica, de las que, tanto usuarios como empresas, están comenzando a sacar partido. De hecho, algunas empresas han construido su modelo del negocio alrededor del uso del software del libre, como hemos visto en este trabajo.

Hemos mostrado ejemplos donde el uso del software libre ha sido el factor clave para hacer de un producto la referencia de un mercado completo, tanto en la investigación como en la educación. En ambos casos, el software libre es más fácil de adaptar a las necesidades particulares de usuarios y reutilizarlo en otras situaciones. Las comunidades del software libre han demostrado que herramientas libres pueden llegar a ser más populares incluso que las proporcionadas por los fabricantes originales.

Finalmente hemos presentado un ejemplo donde el uso del software libre ha permitido alcanzar progresos más rápidos y resultados mejores. Compartiendo las soluciones ya conocidas, los desarrolladores nuevos pueden centrarse en mejorar las partes del sistema que no son está maduras.

6 Agradecimientos

El trabajo de Pablo Barrera ha sido financiado por la Universidad Rey Juan Carlos. El trabajo de Jose M. Cañas, Francisco Martín y Vicente Matellán ha sido financiado parcialmente por el proyecto español ACRAE del MEC, (Dpi2004-07993-c03-01). El trabajo de Gregorio Robles ha sido financiado en parte por la Comisión Europea, bajo CALIBRE CA, el programa de IST, contrato número 004337, y en parte por el Universidad Rey Juan Carlos bajo el proyecto PPR-2004-42.

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>.