

Simuladores de Robo-Futbol y otro Software Libre en la Inteligencia Artificial

Vicente Matellán
vmate@ia.uc3m.es

Camino Fernández
camino@inf.uc3m.es

Departamento de Informática
Universidad Carlos III de Madrid

Resumen

El presente artículo intenta dar una visión de las aportaciones que el software de dominio público puede hacer al campo de la investigación en informática, usando como ejemplo la rama de la Inteligencia Artificial. Para ello se describen las ventajas que aporta el software de libre distribución como filosofía general, a las disciplinas científicas en particular y en especial a las áreas relacionadas con las ciencias de la computación como la Inteligencia Artificial. Para ilustrar algunas de dichas aportaciones se utilizan distintos ejemplos, en especial el de los simuladores de libre distribución de las competiciones de robots autónomos inteligentes en el que trabajan los autores.

1 Introducción

En el campo de la Inteligencia Artificial, en especial en el área de los robots autónomos, existen desde hace años diversas competiciones que tienen como objeto probar la eficiencia de las distintas técnicas de resolución de problemas. Una de las competiciones más llamativas que existen actualmente es la de fútbol entre robots, denominada *RoboCup* [1]. RoboCup es realmente el nombre coloquial de la *World Cup Robot Soccer Initiative*, un intento de promover la investigación en el área de los robots autónomos cooperantes [2].

En paralelo a la competición con robots reales se celebrará una competición basada en un simulador (figura 1) de libre distribución [3]. Esta es nuestra herramienta habitual de trabajo (sobre una plataforma basada totalmente en software de libre distribución) y será, por tanto, el ejemplo que usaremos más frecuentemente en el resto de este artículo para introducir las ventajas, que a nuestro juicio, aporta la filosofía del software de libre distribución a este campo.

En primer lugar parece lógico que si el objetivo es comparar las diferentes arquitecturas, se disponga de una plataforma común que permita la comparación. No sería razonable que cada uno probase su sistema en su propio simulador.

Por otra parte, si el simulador es único, se tiene que garantizar que todos los participantes tengan igualdad de oportunidades en la competición, es decir,



Figura 1: El simulador de la RoboCup

que no haya nadie que, conociendo la estructura del simulador, se aproveche de esa información privilegiada para diseñar su arquitectura.

Una solución podría haber sido garantizar bajo palabra, por escrito o por cualquier otro método tradicional que el código del simulador no sería conocido por ningún participante. Este tipo de soluciones es habitual en muchos otros órdenes de la vida, como los exámenes, concursos públicos, subastas, etc.

Sin embargo, para este caso los organizadores han decidido que existe una solución mejor: hacer público el código del simulador que se empleará. De esta forma se garantiza perfectamente la igualdad de oportunidades y se ayuda a la difusión de la propia competición y del conomiento.

Al permitir que el simulador se distribuya (copie) libremente, se facilita enormemente la difusión de la competición. Si alguien tiene el simulador se lo puede copiar, con toda tranquilidad, a sus compañeros. Tanto es así que se puede conseguir este simulador mediante *ftp* anónimo por Internet o a través de la correspondiente página HTML.

Desde otro punto de vista, se puede decir que con esta política de distribución también se está contribuyendo a difundir el conocimiento, en este caso sobre cómo construir simuladores, a todo el mundo, pues los organizadores facilitan el código. De esa forma cualquier interesado puede leerlo y obtener el conocimiento que busca.

Este ejemplo puede hacer ver que no siempre es conveniente proteger el software con las tradicionales licencias, entregando solamente el código objeto y guardando el fuente como un tesoro y que esto es incluso cierto en situaciones como las competiciones o la ciencia. La pregunta es si esta idea de distribuir los programas con su código, permitiendo además que los que lo adquieran lo redistribuyan libremente (siempre manteniendo los mismos derechos para los siguientes usuarios), como establece, por ejemplo, la licencia de GNU [4], es siempre beneficiosa. Para intentar responderla, en la siguiente sección se analizarán con más detalle los motivos que pueden llevar a considerarlo así.

2 Motivaciones

El caso de la RoboCup no deja de ser un caso particular, que no tiene porqué ser aplicable en todas las ramas de la informática. Por ello conviene analizar qué motivos hacen recomendable el software de libre distribución en los ambientes académicos y de investigación, y cuales desaconsejan su uso.

En primer lugar, hay que tener en cuenta que en gran parte de las actividades humanas, la competitividad parece ser un catalizador eficiente a la hora de aumentar la productividad de las mismas. La ciencia no ha sido, ni es, ajena a este principio. Así, a lo largo de su historia se pueden encontrar multitud de ejemplos de trabajos paralelos y enormemente competitivos entre científicos que perseguían los mismos objetivos, bien con fines militares, comerciales o de prestigio personal. Dentro de esas *carreras* una de las armas tradicionales ha sido el secreto de los trabajos que cada investigador estaba realizando, la discreción sobre las herramientas empleadas y la protección de los resultados obtenidos.

Así mismo, el catalizador de la competencia parece no tener rival a la hora de mejorar el rendimiento en el campo de la investigación y de igual forma las medidas tradicionales de protección de los trabajos de investigación parecen ser estrictamente necesarios para garantizar la rentabilidad de esa competencia.

El mundo de la informática tampoco es ajeno a esta situación. Así, multitud de grupos trabajan actualmente sobre los mismos temas compitiendo, bien por las mismas subvenciones, bien por las mismas cuotas de mercado o de prestigio. De igual forma, las mismas teorías sobre la protección de los resultados mediante licencias y de los métodos empleados (el código fuente de los programas) se ha generalizado en la informática.

Entonces ¿Cuál es la ventaja que aporta el software de libre distribución? ¿No sería más lógico ocultar a los demás el código escrito? ¿No dar ventajas a los posibles competidores, ni pistas sobre las líneas de trabajo? Para muchos desde luego es cierto, pues no permiten que los demás conozcan el código de sus programas, ni permiten que se copien o que se regalen a otro usuario al adquirir una nueva versión, etc.

En el bando opuesto, el colectivo de desarrolladores de software de libre distribución consideran que ese no es el camino adecuado para el caso del software. Para argumentar su opinión se pueden esgrimir variados argumentos, desde los más filosóficos a los más técnicos. Un buen resumen de estos principios, así como

una crítica a la forma cómo las compañías de software están tratando la copia del software se puede encontrar en [5]. Además de las razones esgrimidas en ese artículo, que se puede calificar como manifiesto, en el caso del software de investigación se podrían considerar otros aspectos. A modo de ajemplo citaremos algunos en los siguientes apartados.

2.1 El aspecto social

Un aspecto que debería tener en cuenta cualquier investigador es el alcance social de su trabajo. Los desarrollos que se realizan en los ámbitos académicos, de investigación y en general de innovación, tienen influencia en la vida de todos los ciudadanos, por lo cual la información sobre sus trabajos, descubrimientos y aplicaciones debería ser conocida. Por tanto, los avances científicos deberían mostrar cómo se han obtenido y qué necesitan para mantenerse todos sus trabajos.

De igual forma, los programas deberían poder ser inspeccionados, los usuarios deberían saber qué está haciendo realmente el software que están utilizando. Alguien podría diseñar, por ejemplo, editores de texto que enviasen copias (sin avisar al usuario) de todo lo que se teclea a determinados lugares donde pudiera ser investigado. Este ejemplo, que parece utópico, no lo es tanto si consideramos que el gobierno americano intentó instalar un *chip* en todos los ordenadores que le permitiera intervenir cualquier comunicación electrónica.

Además, hay que considerar que los resultados de la investigación deberían contribuir al desarrollo de la humanidad, mejorando el nivel de vida de las regiones más desfavorecidas y acercándolo al de las más avanzadas. En el caso del software, al permitir que cualquiera pueda acceder al software desarrollado en los lugares más avanzados se está facilitando que los menos desarrollados avancen con mayor rapidez. Este es un arma de doble filo, pues muchos opinan que esta idea favorece a los, en teoría, competidores. No suele ser cierto. Llevándolo a sus límites, poco podrían hacer en Somalia con el software de los transbordadores espaciales, aunque quizá sí les resultase útil algún software de gestión. El problema sería que Japón se aprovechara de los desarrollos americanos, pero eso es también relativamente falso, como se comenta en los siguientes apartados.

2.2 Las ventajas individuales

Siguiendo con la idea del párrafo anterior, hay que ser realistas: la mayor parte de la gente (investigadores incluidos) no son idealistas, buscan su desarrollo personal y como mucho el de su grupo, empresa, nación o en general, organización. Parece ser un sentimiento humano generalizado. Por tanto, no verán con buenos ojos que otros prosperen a costa de su esfuerzo, sin verse ellos recompensados o reconocidos.

En contra de lo que muchos pueden pensar, el software de dominio público también puede ser positivo desde ese punto de vista. ¿Qué mejor publicidad puede tener un programador que decir que su software se utiliza en todo el mundo? ¿Alguien duda de la calidad de un programa como *emacs*? Además,

es bastante evidente que los productos de dominio público hacen bastante más conocidos a los desarrolladores que los productos de las casas comerciales.

En el campo de la investigación esto es todavía más cierto. De hecho, en todos los campos de la ciencia existen multitud de revistas que se dedican precisamente a ello, a hacer **público** el resultado de los trabajos de los científicos. En el caso del software esto es aún más sencillo, pues los desarrolladores pueden generalmente enviar sus resultados (sus programas) por medios electrónicos (Internet) a cualquier colega interesado en probarlos, lo que no es tan fácil en el caso de otras ciencias (como la biología o la química).

Desde otro punto de vista, el de los preocupados por garantizar la autoría de los trabajos científicos, está claro que la mejor forma (ya desde hace siglos) de garantizar la autoría, la primera realización, de un trabajo científico (de un programa en la mayoría de los casos informáticos) es publicándolo en alguna revista del campo. ¿Qué mejor publicación, en el caso informático, que la publicación al mundo entero a través de la Internet? ¿Qué mejor garantía para un científico que el hecho de que existan otros colegas que prueben haber recibido el trabajo (software) directamente de su creador antes que otro pueda afirmar haberlo realizado él?

Pero aún hay más. Muchas empresas invierten fortunas en publicidad tratando de convencer a los usuarios de las bondades de sus productos. Utilizando para ello la técnica del consejo: unos usuarios se los recomiendan a otros que confían en ellos. Del mismo modo ¿Qué garantía es mejor que recibir una copia, idéntica, del producto que está usando el recomendador? Eso es lo que se consigue mediante el software de libre distribución: que los usuarios recomienden con libertad, sin tener que auto-justificarse, los productos que les parezcan adecuados.

El mismo mecanismo es aplicable al mundo de la investigación, en particular a la informática y dentro de ella a la Inteligencia Artificial. La mejor forma de mostrar que el sistema que se ha desarrollado resuelve los problemas que alguien afirma que resuelve y de explicar cómo lo hace es, a la vez que contarlos literariamente, hacerlo mediante el código que implementa esa idea. Pudiéndose verificar si la solución realmente funciona, si es escalable para otros problemas, etc.

Por último, no se puede dejar de citar el enorme *feed-back* que en el caso del software de libre distribución se obtiene de los usuarios. En general, los mejores depuradores de un programa son los usuarios del mismo. Además, en el caso de usuarios avezados, como suele ser el caso de los investigadores en Inteligencia Artificial, si disponen del código pueden hacer sugerencias, comunicar mejoras, etc. al creador del software. De hecho, para los paquetes que se califican como de libre distribución se suele crear una lista de correo que facilite la comunicación entre los usuarios de ese software. Por ejemplo, en el caso de la RoboCup existe la lista RoboCup@csl.sony.co.jp, a la que cualquiera puede suscribirse, en la que se anuncian las nuevas versiones, se comentan los errores detectados o se hacen consideraciones sobre las nuevas mejoras o modificaciones que se deberían incorporar al simulador.

2.3 El realismo práctico

Los que aún tengan reparos en hacer público su software, pueden asumirlo pensando que la mayor parte de los programadores que dejan su software libre lo hacen cuando consideran que el producto está terminado, lo cual generalmente quiere decir que ya están trabajando en otro tema, o en otra versión.

Traducido al campo de la investigación quiere decir que no están *regalando* su último trabajo. Los investigadores, en general, hacen público lo que ya tienen terminado, no aquello en lo que están trabajando. De esta forma, cualquiera tiene todo el derecho a fijar el ritmo con el que va dejando al resto acceder a su información.

Este es el caso de los autores que deciden depositar sus aplicaciones en el repositorio de inteligencia artificial de la Universidad Carnegie Mellon. El software contenido en este lugar se distribuye bajo las condiciones de libre redistribución, libre uso y uso bajo la propia responsabilidad, más otras condiciones que impongan los propios autores siempre que no contradigan las anteriores, como GNU [4] o restricciones a su uso comercial, militar, etc.

3 Conclusiones

En las secciones anteriores se ha tratado de explicar qué entienden los autores como ventajas del software de libre distribución en el campo de la investigación en general y en particular en la Inteligencia Artificial. Además de las justificaciones genéricas se han comentado casos particulares, como el de la RoboCup, donde se pueden apreciar muchas de estas ventajas. En cualquier caso, la única conclusión que se debería sacar de este artículo es que existe una forma alternativa de trabajar en el mundo del software, incluido el software de investigación, al copyright tradicional. La elección de la filosofía de trabajo debe ser personal, pero tomada desde la reflexión y el conocimiento.

Por nuestra parte, hemos aceptado desarrollar software de libre distribución, siempre que sea posible. De esta forma haremos públicas aquellas herramientas que creamos que pueden interesar a alguien más. Como ejemplo, sirva citar el simulador distribuido de robots que estamos desarrollando en el Laboratorio de Agentes Inteligentes [6], que esperamos dejar como de libre distribución tan pronto como esté terminado. Esa creemos que es la forma de progresar y de hacerlo además en la dirección correcta, gracias a las opiniones de los demás.

4 Direcciones de interés

Como ya se ha comentado, el software de libre distribución tiene, entre otras ventajas, la facilidad para obtener todo tipo de programas a través de Internet, incluso de forma anónima al no existir ningún problema de distribución.

A continuación se encuentran las direcciones del software, por ejemplo sobre simuladores de fútbol entre robots, mencionados en este trabajo, alguna sobre

software de dominio público para la inteligencia artificial y sobre el software libre en general:

<http://www.robocup.org/RoboCup/RoboCup.html>: Página oficial de la RoboCup. En ella se pueden obtener las normas de la competición y el simulador.

<http://wwwi3s.unice.fr/~om>: Olivier Mitchel organiza un concurso basado en el mini-robot Khepera (información de este robot comercial se puede encontrar en: <http://www.epfl.ch/>). A través de la página de Olivier se puede obtener la última versión de su simulador.

<http://www.mirosot.org/MIROSOT96/index.html>: Información sobre los encuentros previos de la RoboCup.

<http://forum.swarthmore.edu/~jay/learn-game/projects/microb.html>: Otra versión del fútbol entre robots: MICROB. En este caso es una versión francesa, donde también han desarrollado su propio simulador.

<http://www.cs.ubc.ca/nest/lci/soccer>: EL LCI Dynamite Testbed. Uno de los primeros entornos, en este caso canadiense, para la competición entre robots [7]. También tienen su propio simulador.

<ftp://irisa.irisa.fr/pub/gnu/>: Dirección europea donde se puede conseguir software de libre distribución de la Free Software Foundation, como emacs, gcc, etc.

<http://www.inf.uc3m.es> : Página HTML del Departamento de Informática de la Universidad Carlos III de Madrid, donde se puede obtener más información sobre los autores.

<http://www.ptf.com/ptf/products/AI> : Repositorio de Inteligencia Artificial de la Universidad Carnegie Mellon. Contiene implementaciones de lenguajes, software diverso, tesis, libros, anuncios de conferencias, etc.

Referencias

- [1] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. Robocup: The robot world cup initiative. In *Proceedings of the IJCAI-95 Workshop on Entertainment and AI/Life*, pages 19–24, 1995.
- [2] Third call for participation. Micro-robot world cup soccer tournament, 1996.
- [3] Itsuki Noda. Soccer server: A simulator of robocup. In *Proceedings of AI Symposium'95*. Japanese Society for Artificial Intelligence, December 1995.
- [4] Free Software Foundation. Gnu general public license, 1991.

- [5] Richard Stallman. Why software should not have owners. In *Comunicaciones del Primer Taller sobre el Software de Libre Distribución*, pages 13–17, Septiembre 1995.
- [6] Lorenzo Sommaruga, Ignacio Merino, Vicente Matellán, and José Manuel Molina. A distributed simulator for intelligent autonomous robots. In *Proceedings of the Fourth International Symposium on Intelligent Robotic Systems*, Lisbon, (Portugal), 1996.
- [7] Michael K. Sahota, Alan K. Mackworth, Rod A. Barman, and Stewart J. Kingdon. Real-time control of soccer-playing robots using off-board vision: the dynamite testbed. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 3690–3663, 1995.