

GSI-Robocup.

Un entorno de enseñanza e investigación de Agentes Inteligentes basado en Programación con Reglas

Carlos A. Iglesias

Grupo de Sistemas Inteligentes
Depto. de Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid
E-mail: cif@dit.upm.es

Mercedes Garijo

Grupo de Sistemas Inteligentes
Depto. de Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid
E-mail: mga@dit.upm.es

Resumen—En este trabajo se presenta un entorno diseñado para la enseñanza de agentes inteligentes en la escuela de ingeniería de telecomunicación. El diseño del entorno permite que los alumnos programen, con el motor de reglas Jess, agentes que juegan al fútbol en la competición Robocup. La programación basada en reglas en un entorno basado en un dominio real facilita la comprensión y reutilización del conocimiento de los agentes. El disponer de un dominio atractivo favorece el interés y motivación de los alumnos.

Index Terms—Sistemas inteligentes, robocup, reglas, ingeniería telecomunicación.

I. INTRODUCCIÓN

Este artículo presenta nuestra experiencia en el uso de la *Robocup* para la enseñanza de Sistemas Inteligentes en Ingeniería, y el entorno *GSI-Robocup* que hemos desarrollado para combinar la enseñanza de agentes inteligentes con la programación declarativa basada en reglas.

El resto del artículo se organiza como sigue. En la sección II se realiza una introducción a la competición Robocup. A continuación, en la sección II se presenta el enfoque seguido en la aplicación de la competición Robocup a la enseñanza de sistemas inteligentes en Ingeniería de Telecomunicación, y en la sección IV, el entorno desarrollado para su programación mediante reglas. Por último, se presentan las conclusiones y trabajos futuros en la sección VI.

II. LA COMPETICIÓN ROBOCUP

La competición Robocup [1] es un proyecto internacional para fomentar la investigación en inteligencia artificial, especialmente en robótica y agentes inteligentes. Anualmente, desde 1997, se celebra una competición mundial en la que pueden participar, en diferentes modalidades y categorías, grupos de investigación que hayan desarrollado equipos de robots inteligentes, físicos o simulados, capaces de jugar al fútbol. En la modalidad de simulación, los equipos están formados por agentes software, cada uno de los cuales simula el comportamiento de un jugador de fútbol. Dentro de esta modalidad, hay dos ligas distintas para entornos de agentes simulados en dos o en tres dimensiones. Este artículo se centra

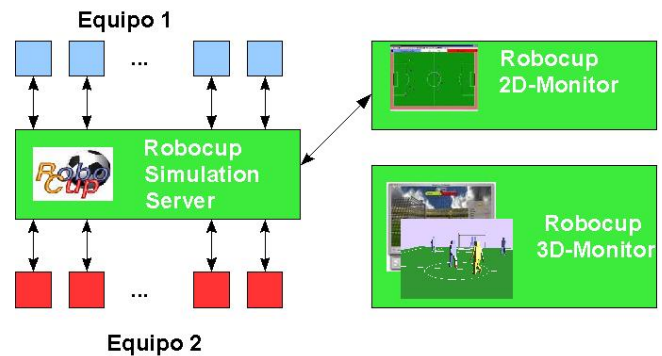


Figura 1. Arquitectura del Simulador Robocup

en el desarrollo de sistemas inteligentes capaces de competir en la liga de simulación en dos dimensiones. El simulador de la competición Robocup es un sistema distribuido cuya arquitectura se muestra en la figura 1. El elemento central es el simulador (*robocup soccer server*) [2], que actúa como servidor y realiza la simulación del juego. Junto con el servidor, se distribuye un programa que visualiza el juego en dos dimensiones (*robocup soccer monitor*), aunque existen desarrollos disponibles de otros monitores en tres dimensiones.

La comunicación entre el simulador y los clientes se realiza mediante sockets UDP/IP, lo que facilita que los jugadores puedan ser programados en cualquier lenguaje. Los jugadores de los equipos actúan como clientes que, una vez conectados al servidor, reciben información sensorial y mandan órdenes para controlar a cada futbolista (avanzar una distancia en una dirección, chutar con una determinada fuerza, etc.).

El simulador permite emplear un árbitro automático o manual, y define reglas sobre la duración del partido, fuera de juego, o número de partes y duración de cada una. Además, se definen normas de conducta, que impiden el envío masivo de órdenes al servidor

Los agentes disponen de tres tipos de sensores: auditivo, visual y corporal. El *sensor auditivo* permite escuchar los mensajes del entrenador, del árbitro y de los otros jugadores de

su mismo equipo. El simulador limita la capacidad auditiva de los jugadores para evitar un uso excesivo de este mecanismo, y los mensajes sólo se reciben si ambos están a una distancia menor de un umbral. El *sensor visual* informa periódicamente sobre los objetos que ve el jugador y su posición respecto del jugador. Los objetos pueden ser la portería, las líneas, otros jugadores y el balón. El jugador también cuenta con un área de visión que depende de su posición en el campo y la posición de su cabeza. Con el fin de introducir ruido, se introduce imprecisión sobre la posición de los objetos lejanos. El *sensor corporal* proporciona información sobre el estado físico del jugador, que incluye la calidad y anchura de visión, la velocidad y la dirección actual, la dirección de la cabeza respecto del cuerpo y la potencia del jugador. El simulador decreta esta potencia a medida que el jugador hace ejercicio físico, lo que limita su velocidad. Con el fin de incluir el movimiento inesperado de objetos como sucede en el mundo real, se introduce ruido en el modelo de movimiento de los objetos. Los jugadores pueden acelerar hacia una posición, chutar, girar el cuerpo o la cabeza, o mandar mensajes. El portero es un jugador especial que puede *coger* la pelota, y dispone de parámetros de simulación específicos.

III. APLICACIÓN DE LA COMPETICIÓN ROBOCUP EN EDUCACIÓN

Como vemos, la competición Robocup ofrece un entorno de experimentación muy interesante. El simulador ofrece un entorno de simulación con la complejidad de recibir y actuar en *tiempo real* y, además, la *toma de decisiones es distribuida*, y nos ofrece facilidades para la *coordinación* en un entorno competitivo.

El hecho de abordar un dominio tan popular y atractivo como el fútbol, ha facilitado que se aplique a la docencia de sistemas inteligentes y robótica ([3], [4]). Sin embargo, el modelo del servidor la competición Robocup es complejo para que los alumnos puedan desarrollar prácticas. Entre las dificultades a las que los alumnos se tienen que enfrentar están las siguientes:

- Requiere programación de la conexión con el servidor, y gestión de sockets UDP.
- Requiere realizar un programa complejo para analizar los mensajes que envía el servidor a los jugadores.
- Requiere programar cómo se modelan y actualizan los objetos del campo.
- El modelo de posiciones relativas de la competición Robocup es complejo para razonar directamente sobre él, y requiere realizar un procesamiento geométrico de la información recibida.
- Al tratarse de un simulador en tiempo real, es fácil hacer implementaciones en que un agente pierda mensajes, o los jugadores empleen información no actualizada.

En nuestro caso, además, queríamos que este dominio nos sirviera para reforzar la programación declarativa basada en reglas, por ser habitualmente una de las mayores dificultades encontradas por los alumnos, que están habituados a la programación imperativa.



Figura 2. Arquitectura de los clientes GSI-Robocup

Con este fin, hemos desarrollado una biblioteca, GSI-Robocup, que se presenta en la sección IV, que facilita el desarrollo de sistemas inteligentes capaces de competir en la liga de simulación en dos dimensiones, reduciendo el desarrollo de los agentes a la programación de las reglas de comportamiento.

IV. LA BIBLIOTECA GSI-ROBOCUP

La biblioteca GSI-Robocup ([5], [6], [7], [8]) se encarga de gestionar toda la comunicación con el servidor, liberando al programador de las labores de bajo nivel y permitiéndole centrarse en la programación del comportamiento *inteligente* del jugador utilizando el lenguaje de reglas Jess [9].

La biblioteca GSI-Robocup presenta una arquitectura con una capa Java que integra los planificadores escritos en Jess, tal como se muestra en la figura 2.

La capa Java ofrece las siguientes funcionalidades:

- Gestión de comunicaciones UDP
- Análisis de mensajes recibidos del servidor, actualización de los objetos conocidos por el agente, y obtención de posiciones absolutas de los objetos mediante cálculos geométricos.
- Gestión de hebras para recibir y procesar mensajes, de forma que no se pierdan mensajes.
- Gestión de hebras para ejecutar los jugadores. Aunque los agentes pueden lanzarse como procesos independientes, se lanzan en hebras con el fin de que supongan una menor carga.

La biblioteca GSI-Robocup permite que los alumnos se centren en programar los planificadores en Jess, y se liberan de la problemática de conexión con el servidor. Esta capa actualiza la memoria de trabajo con la información recibida. A efecto ilustrativo, en el listado 1 se muestra un ejemplo de regla para avanzar hacia el balón, que indica que si el juego está en activo (en modo *play_on*), y el jugador no ha visto el balón desde hace 5 ciclos, y el balón está lejos (a más de 1 metro), debe girarse en la dirección de la pelota, y correr con la máxima potencia (al 100 %).

Programa 1 Ejemplo de planificador

```
(defrule runToBall
  (playMode (realPlayMode "play_on"))
  (playerData (dontSeeBallTime ?t)
    (id myself)
  )
  (ball (distance ?ballDistance)
    (direction ?ballDirection)
  )
  (test (and (< ?t 5)
    (> ?ballDistance 1)
  )
  )
)
=>
  (turn ?ballDirection)
  (dash 100.0)
)
```

V. EXPERIENCIA DOCENTE CON GSI-ROBOCUP

La biblioteca GSI-Robocup viene siendo usada en la E.T.S.I de Telecomunicación de la Universidad Politécnica de Madrid desde 2001 en la asignatura *Laboratorio de Sistemas Inteligentes*. En concreto, se emplea en la última práctica, que es la más extensa. En esta práctica los alumnos deben desarrollar un sistema en equipo, y pueden escoger entre varias alternativas de desarrollo de un sistema multiagente. La GSI-Robocup ha sido seleccionada todos los años por al menos un tercio de los grupos, a pesar de ser una práctica que requiere gran dedicación.

Desde que se utiliza esta biblioteca, los grupos de alumnos desarrollan equipos de jugadores de fútbol cuyos comportamientos son notablemente más *inteligentes* que los que se conseguían sin la biblioteca e incluso, en muchas ocasiones superan nuestras expectativas. Los alumnos son capaces de desarrollar equipos en los que hay agentes con roles de conducta bien diferenciados (delantero, defensa, portero, ...) y también desarrollan estrategias de equipo (más agresivo si estoy perdiendo, dar pases entre compañeros, ...). Al centrarse únicamente en la programación de reglas y disponer de los desarrollos realizados por alumnos de cursos anteriores, los avances en la mejora de los comportamientos según su rol, su posición, su estrategia, etc. son realmente visibles, especialmente cuando en las demostraciones cuando se realizan competiciones entre equipos desarrollados por distintos grupos o contra equipos desarrollados en cursos anteriores. Por otra parte, y esto también es muy importante, con la realización de este tipo de prácticas diferentes se advierte una mejora notable en el estilo de programación, dado que las reglas deben procesar los eventos del servidor, y no es posible *cablear* el control de la aplicación.

Las principales dificultades encontradas por los alumnos han sido:

- Dificultades en la depuración. Para muchos alumnos, este proyecto es su primer contacto con la programación de

sistemas distribuidos, sin un log centralizado. Si bien la competición Robocup ofrece un componente *logPlayer* que permite la reproducción de jugadas grabadas, es necesario observar por qué se activan o no las reglas. A esta dificultad, sumamos el hecho del ruido introducido por el simulador.

- Dificultades con el procesamiento de los mensajes. El análisis de los mensajes del servidor es complejo y consumía mucho tiempo, lo que hacía que se perdieran mensajes.
- Elevada carga de los agentes.

Para superar estos problemas, este año se ha rediseñado la plataforma, optimizando el procesado de los mensajes, e implementando que los agentes se gestionen como hebras. Se han realizado pruebas extensivas con JUnit y fuerte carga con el fin de reducir la memoria y capacidad de procesamiento requerido. Además, con el fin de acelerar el aprendizaje, se han elaborado tutoriales en la herramienta de e-learning Moodle, y se les proporcionan a los alumnos equipos anteriores, para que puedan evolucionar soluciones anteriores.

VI. CONCLUSIONES Y TRABAJOS FUTUROS

Nuestras experiencias con la biblioteca GSI-Robocup han sido muy satisfactorias y han mostrado que, además de que el dominio motiva enormemente a los alumnos, especialmente a los aficionados al fútbol, que son una amplia mayoría, la incorporación de nuestra biblioteca facilita tanto la programación de los jugadores que permite avances significativos en los resultados, perfectamente visibles en las demostraciones realizadas por los alumnos.

Desde el punto de vista pedagógico, el dominio de la competición Robocup permite enseñar las principales bases de los agentes inteligentes, como son las capacidades de percepción y actuación con el entorno, la comunicación con otros agentes, y la planificación personal de las acciones. En cuanto a la coordinación entre agentes, es habitual que se pretenda hacer una coordinación explícita basada en comunicación directa pero, gracias a que esta opción está penalizada por el simulador, se tiende a una coordinación basada en roles (el esquema de juego).

Este año se ha trabajado sobre todo en mejorar la documentación y crear ejemplos y tutoriales para acelerar el aprendizaje de la plataforma, y en optimizarla. Entre los trabajos futuros, se está trabajando en la creación de una biblioteca que permita almacenar las conductas creadas por los alumnos cada año, y que permita hacer competiciones online, ejecutando los equipos y dando los resultados de la liga.

AGRADECIMIENTOS

Los autores quieren agradecer a todos los alumnos que han colaborado en el desarrollo de la plataforma, Álvaro Martínez, Roberto Somolinos, Luis Manuel Cruza, Víctor Sánchez y Francisco Santiago, así como a los alumnos de *Laboratorio de Sistemas Inteligentes* por sus sugerencias durante estos años.

REFERENCIAS

- [1] The Robocup Federation, "Web oficial de la competición robocup." [Online]. Available: <http://www.robocup.org>
- [2] M. Chen, E. Foroughi, F. Heintz, Z. Huang, K. K. Spiros Kapetanakis, J. Kummeneje, I. Noda, O. Obst, P. Riley, T. Steffens, Y. Wang, and X. Yin, "Users manual. robocup soccer server for soccer server version 7.07 and later," The Robocup Federation, Tech. Rep., Aug. 2002, available at <http://server.sourceforge.net/docs/manual.pdf>.
- [3] J. Vidal and P. Buhler, "Teaching multiagent systems using RoboCup and biter," *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, vol. 4, no. 2, 2002. [Online]. Available: citeseer.ist.psu.edu/vidal02teaching.html
- [4] E. Sklar, S. Parsons, and P. Stone, "Robocup in higher education: A preliminary report." in *RoboCup*, ser. Lecture Notes in Computer Science, D. Polani, B. Browning, A. Bonarini, and K. Yoshida, Eds., vol. 3020. Springer, 2003, pp. 296–307.
- [5] Grupo de Sistemas Inteligentes (GSI), Departamento de Ingeniería de Sistemas Telemáticos, E.T.S.I. Telecomunicación, Universidad Politécnica de Madrid, "Web oficial de gsi robocup." [Online]. Available: <http://www.gsi.dit.upm.es/robocup>
- [6] Álvaro Martínez Reol, "Desarrollo de un sistema multiagente integrando jess y java para la robocup," Master's thesis, E.T.S.I. Telecomunicación, Univ. de Valladolid, 2000.
- [7] R. Somolinos, "Desarrollo de un entorno java-jess de composición de habilidades de agentes inteligentes para la competición robocup," Master's thesis, E.T.S.I. Telecomunicación, Univ. Politécnica de Madrid, 2003.
- [8] L. M. Cruza, "Aplicación web de composición de agentes inteligentes para la competición robocup," Master's thesis, E.T.S.I. Telecomunicación, Univ. Politécnica de Madrid, 2006.
- [9] Sandia National Laboratories, "Web de Jess. The Rule Engine for the Java Platform." [Online]. Available: <http://www.jessrules.com/jess/index.shtml>