

# Diseño de Agentes experimentando con robots que juegan al Fútbol en ambientes reales y simulados\*

Pablo Kogan                      Gerardo A. Parra                      Rodolfo Del Castillo  
pkogan@uncoma.edu.ar    gparra@uncoma.edu.ar    rolo@uncoma.edu.ar

Grupo de Investigación en Robótica Inteligente  
Departamento de Ciencias de la Computación  
Facultad de Economía y Administración  
Universidad Nacional del Comahue  
Buenos Aires 1400 - 8300 Neuquén - Argentina  
Tel/Fax (54) (299) 4490312/313

## Resumen

El objetivo del Grupo de Investigación en “Robótica Inteligente”, es promover el estudio de áreas de Inteligencia Artificial a través del fútbol con robots. En este artículo se describe el diseño de un agente en Prolog, logrando una abstracción de la plataforma y tipos de robots que se utilicen. Para evaluar el diseño, se ha experimentado sobre una plataforma real y otra simulada, utilizando el mismo agente lógico.

**PALABRAS CLAVES:** Sistemas Multiagentes. Agentes Inteligentes. Robótica. Programación Lógica. Fútbol con robots.

## 1. Introducción

El fútbol con robots ha sido recientemente catalogado como un problema estándar para la investigación en Inteligencia Artificial y Robótica. Un problema estándar provoca que gran cantidad de investigadores se enfoquen sobre los mismos temas, de forma tal que, aunando esfuerzos el avance sea mucho mayor

que si cada uno trabajara en temas no relacionados.

El fútbol con robots fue elegido para encontrar necesidades complejas del mundo real, a través de un mundo limitado, que tiene problemas manejables en tamaño y costos de investigación. El problema es atractivo porque ofrece una integración de áreas de Inteligencia Artificial y Robótica. Tales áreas incluyen: comportamiento reactivo, adquisición de estrategias, aprendizaje (learning), planeamiento (planning) en tiempo real, sistemas multiagentes, reconocimiento del ambiente, visión, control de motores, control de robot inteligentes, entre otros [1].

Trabajar con robots que jueguen al fútbol, es interesante, por lo que este juego representa a nivel mundial. El fútbol es idioma universal, sus reglas son de conocimiento común y es el deporte más popular en la mayoría de los países. Esta característica hace al problema del fútbol con robots mucho más atractivo para incentivar estudiantes, investigadores, sponsors y a la sociedad en su conjunto. En [4], se presenta el desarrollo de una interfaz entre Prolog y un simulador de fútbol con robots [8], con el fin de participar de la competencia CAFR 2005 [7]. En el presente trabajo se desarrolla una interfaz análoga, pero con el objetivo de que el comportamiento sea igual al de la interfaz con el ambiente real. De esta manera se abstrae el diseño de los agentes, de la plataforma y tipos de robots que se estén utilizando. A continuación, se rea-

---

\*Este trabajo está parcialmente financiado por la Universidad Nacional del Comahue, en el contexto del Proyecto de Investigación “Técnicas de Inteligencia Computacional para el diseño e Implementación de Sistemas Multiagentes” (COD 04/E062), por el Grupo de Investigación en Robótica Inteligente y por la Universidad Politécnica de Madrid a través del Proyecto AL05.PID.0040, “Implementaciones y Modelos de Razonamiento basado en Programación Lógica”.

liza un análisis del diseño de un agente que juega al fútbol en Prolog [11] y se comentan las experiencias realizadas sobre robots reales [9] y sobre el ambiente simulado [8].

## 2. Diseño de un Agente en Prolog

El enfoque tradicional de la construcción de agentes, está dado por la representación simbólica de su ambiente, de su comportamiento y la manipulación sintáctica de esta representación. La arquitectura de agentes basada en lógica utiliza el enfoque tradicional, realizando las representaciones como fórmulas lógicas y la manipulación sintáctica como deducción lógica o demostración de teoremas [2].

En este trabajo, para el desarrollo del agente, se ha seleccionado el lenguaje Prolog, debido a su utilidad como herramienta de representación de conocimiento y razonamiento. El intérprete utilizado es el Ciao Prolog [11], por su facilidad de interacción con el lenguaje C, para trabajar a bajo nivel.

La representación del ambiente del fútbol con robots está dada por la ubicación y orientación de cada uno de los jugadores, la ubicación de la pelota y los límites de la cancha. Por ejemplo, una posible representación del ambiente a través de reglas puede ser:

```

limites_cancha(inferior(0,0), superior(90,50)).
arco_propio(inferior(90,20), superior(90,40)).
arco_contrario(inferior(0,20), superior(0,40)).
pelota(posicion(4,4)).
arquero_propio(posicion(88,25),90).
arquero_contrario(posicion(2,20),180).
jugador1_propio(posicion(15,10),45).
jugador1_contrario(posicion(3,5),315).

```

La base de conocimiento es toda la información que el agente tiene acerca del ambiente. Esta información puede ser errónea porque fallen los sensores o porque es información antigua en el momento de ejecutar la acción. La base de conocimiento puede mantener un registro de las posiciones anteriores de los jugadores y la pelota, de forma tal de predecir futuros movimientos o detectar fallos en los sensores.

La teoría de agentes define que un agente es una entidad que percibe del ambiente a través de sensores y actúa en consecuencia a través de efectores. Una forma de representar el funcionamiento de un agente en Prolog es la siguiente:

```

ciclo : -
    percibir,
    razonar(Accion),
    actuar(Accion),
    ciclo.

```

El predicado *ciclo* representa el ciclo percibir, razonar y actuar, a través de las reglas que lo conforman. La regla *percibir* modifica las reglas de la base de conocimiento para que queden consistentes con la información del ambiente que es percibida a través de los sensores. La regla *razonar(Accion)* realiza el razonamiento de agente, resolviendo la Acción a tomar en base al estado actual de la base de conocimiento. Finalmente la regla *actuar(Accion)* envía los comandos necesarios a los efectores, para que el agente pueda realizar la Acción resuelta. Luego comienza un nuevo ciclo cuando se vuelve a ejecutar la regla percibir. La abstracción con los sensores y efectores, lograda a través de las reglas *percibir* y *actuar(Accion)*, permite que el razonamiento del agente sea el mismo para diferentes ambientes y tipos de robots. Un ejemplo del comportamiento del arquero *kiñe*<sup>1</sup>, está representado en el predicado *razonar*:

```

razonar(Accion) : -
    pelota_en_area - > despejar_pelota(Accion);
    pelota_dir_area - > esperar_en_area(Accion);
    ir_a_posicion_base(Accion).

```

La estrategia del arquero es la siguiente: si la pelota se encuentra en el área del arco propio, entonces despeja la pelota; si la pelota va en dirección al área, entonces espera la pelota dentro del área, en la posición a la que se dirige esta; caso contrario se dirige al centro del arco.

La representación del comportamiento está dada por las instrucciones de movimiento para los robots. Las instrucciones

<sup>1</sup>En lengua Mapuche, número uno.

de movimiento presentan diferente nivel de abstracción que van desde primitivas de velocidad en cada motor del robot, hasta primitivas de más alto nivel como ir a una determinada posición o patear la pelota.

El predicado *razonar* está formado por reglas de la forma *estado* → *actuar*. El predicado *estado* está condicionado por la información que el agente tiene acerca del ambiente. El predicado *actuar* está formado por reglas de la misma forma que el predicado *razonar*, con el fin de ir refinando las acciones hasta llegar a las primitivas básicas necesarias para ejecución de la acción [4].

### 3. Plataforma real

La plataforma utilizada para experimentar con robots reales, está basada en una de las categorías (E-League) de la competencia RoboCup[5]. La categoría tiene software de visión y de comunicación estándar para el funcionamiento de los robots.

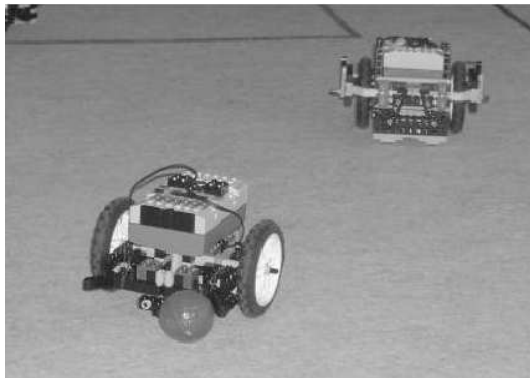


Figura 1: Plataforma real, robots LEGO Mindstorms

Una cámara de vídeo captura imágenes del campo de juego, que son enviadas a un servidor de vídeo llamado Doraemon [10]. Este programa procesa la imagen, extrayendo información del campo de juego como la posición y el ángulo de orientación de cada jugador y la posición de la pelota. Esta información es empaquetada y enviada a cada uno de los dos equipos. El protocolo de comunicación sigue el modelo cliente-servidor de la siguiente manera: los agentes son clientes que reciben del servidor de vídeo paquetes con la

información del ambiente, a través de sockets udp.

El funcionamiento de los efectores también sigue el modelo cliente-servidor, en donde los agentes son clientes de un servidor de comandos. En cada ciclo los agentes le envían al servidor de comandos, paquetes con las primitivas básicas de movimiento, a través de sockets udp. El servidor de comandos se encarga de enviar esta información a los robots [9], a través de transmisores infrarrojos [3]. En el agente diseñado, el predicado *percibir* es el encargado de procesar el paquete enviado por el servidor de vídeo, para modificar la base de conocimiento y el predicado *actuar*(*Accion*) es el encargado de enviar la información de la Acción seleccionada al servidor de comandos.

### 4. Plataforma simulada

Se ha experimentado con el simulador Robot Soccer v1.5a [8], el cual es utilizado en la Categoría “Middle League SimuroSot” del Campeonato Mundial de Fútbol de Robots FIRA [6] y en el Campeonato Argentino de Fútbol de Robots CAFR [7].

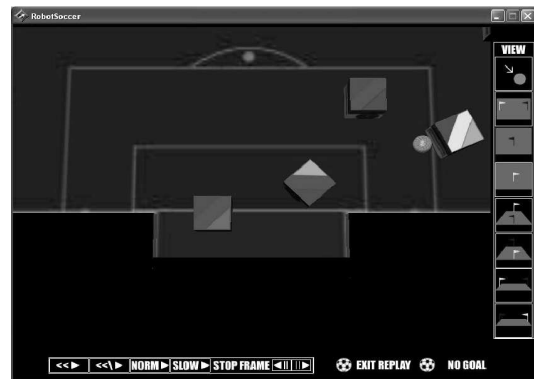


Figura 2: Plataforma simulada, SimuroSot Robot Soccer v1.5a

Se elige inicialmente esta plataforma simulada por su facilidad de uso, para independizarse del robot real y por ser la utilizada en la única competencia que se realiza a nivel nacional.

Por defecto, las estrategias de cada equipo pueden ser desarrolladas en Lingo o una dll (Dynamic Link Library) en C++. Por esta

razón para programar el agente en Prolog, se realizó una interfaz entre C++ y Prolog a través de sockets udp.

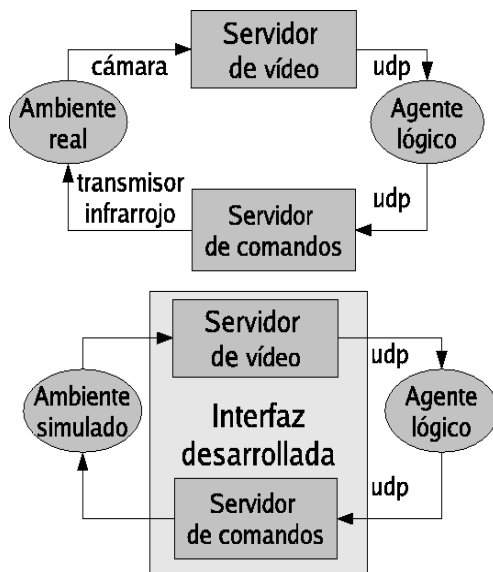


Figura 3: En la figura se describe claramente el objetivo del presente trabajo, el cual es abstraer el agente lógico de la plataforma y tipo de robots que se utilicen.

En este trabajo se desarrolla una interfaz con el objetivo, de que el comportamiento sea igual al de la interfaz con el servidor de vídeo y servidor de comandos de la plataforma real. De esta forma se logra que el mismo agente lógico funcione sobre la plataforma real y la plataforma simulada, como muestra la Figura 3.

El simulador llama a una función de la dll 60 veces por segundo. Esta función decodifica toda la información del ambiente y la envía a través de un socket udp emulando al servidor de vídeo. Luego, a través de otro socket udp, la función espera recibir la acción tomada, emulado al servidor de comandos. Una vez recibida la acción, es decodificada y enviada al simulador para que se muevan los robots. En el agente diseñado hay que modificar el predicado percibir para que reciba la información del emulador del servidor de vídeo y el predicado *actuar(Accion)* para que envíe la información al emulador del servidor de comandos.

## 5. Conclusiones y Trabajos Futuros

La abstracción del agente, en la interacción con el ambiente permitió, que el mismo agente lógico funcione sobre la plataforma real y la plataforma simulada. Esta característica nos permite experimentar diferentes diseños de agentes sobre cada una de las plataformas, sin modificar el agente lógico.

En esta primera etapa, los agentes muestran un comportamiento básicamente reactivo. Se espera mejorar el desarrollo de los agentes, con aportes de áreas como learning, planning y modelado de oponentes. Uno de los objetivos de este trabajo es participar en alguna de las competencias, por lo cual se debe mejorar en la estrategia del equipo y las primitivas de movimiento, para lograr un buen rendimiento.

Otro de los objetivos es trabajar en conjunto con la cátedra de Inteligencia Artificial, utilizando la interfaz desarrollada entre Prolog y el simulador de fútbol con robots, en la práctica del próximo cursado de la materia. Concretamente, los alumnos aplicarán las técnicas aprendidas durante el cursado, en prácticas de laboratorio que consistirán en el desarrollo de agentes inteligentes que asuman alguno de los roles del equipo dentro del campo de juego.

## Referencias

- [1] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa; *RoboCup: The Robot World Cup Initiative*, 1995.
- [2] Michael Wooldrige; *Intelligent Agent; Multiagent Systems A Modern Approach to Distributed Artificial Intelligence* edited by Gerhard Weiss, 1999.
- [3] Alejandro J. Garcia, Gerardo I. Simari, Telma Delladio, Diego R. Garcia, Mariano Tucat, Nicolás D. Rotstein, Fernando A. Martin, Sebastián Gottifredi; *Cognitive Robotics in a Soccer Game Domain: a Proposal for the E-League Competition*, WICC 2004.

- [4] Mauro J. Gómez, José H. Moyano, Nicolás D. Rotstein, Telma Delladio, Alejandro J. García; *LogBall: un equipo de fútbol implementado como un sistema heterogéneo de múltiples agentes lógicos*, WCAFR 2005.
- [5] RoboCup. <http://www.robocup.org>
- [6] FIRA (Federation of International Robot-soccer Association). <http://www.fira.net>
- [7] CAFR (Campeonato Argentino de Fútbol con Robots).  
<http://www.unimoron.edu.ar/cafr2005>
- [8] Robot Soccer Simulator v1.5a.  
[http://www.fira.net/soccer/simurosot/R\\_Soccer\\_v15a.030204.exe](http://www.fira.net/soccer/simurosot/R_Soccer_v15a.030204.exe)
- [9] LEGO Mindstorms.  
<http://www.legomindstorms.com>
- [10] Doraemon.  
<http://sourceforge.net/projects/robocup-video>.
- [11] Ciao Prolog.  
<http://clip.dia.fi.upm.es/Software/Ciao/>