

JGOMAS: Una alternativa a las prácticas de Sistemas Inteligentes

Antonio Barella, Soledad Valero, Carlos Carrascosa
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
e-mail: {tbarella, svalero, carrasco}@dsic.upv.es

Resumen

Con el fin de hacer más atractivas las técnicas de Inteligencia Artificial a los alumnos, así como facilitar a los alumnos su contacto con herramientas actuales y ampliamente utilizadas tanto en investigación como en desarrollo, se han realizado una serie de cambios en el contenido de las prácticas de la asignatura *Sistemas Inteligentes* (optativa de quinto curso de II). Para ello, se ha creado un entorno diferente para la realización de las prácticas. Éste está compuesto por una herramienta de desarrollo y ejecución de agentes, denominada JGOMAS, así como una Web¹ dedicada a la misma, donde los alumnos pueden acceder a diversa documentación e información, así como interactuar con el profesorado.

1. Introducción

En el último curso de la titulación de Ingeniería Informática de la Facultad de Informática de Valencia se imparte una asignatura optativa denominada **Sistemas Inteligentes (SIN)**. A lo largo de este artículo presentaremos el nuevo planteamiento de las prácticas de dicha asignatura, llevado a cabo para ofrecer una visión actualizada de los avances de las técnicas objeto de estudio de la asignatura, así como mantener una alta motivación entre los alumnos.

Esta asignatura, junto con **Técnicas de Inteligencia Artificial (TIA)** también optativa del mismo curso y cuatrimestre, viene a complementar la formación del alumno sobre las diversas técnicas de Inteligencia Artificial (introducidas en una asignatura previa en cuarto curso), presentando el uso y demostración de un conjunto de técnicas y herramientas actuales. Las técnicas estudiadas deben ser contrastadas y

evaluadas aplicándolas en la resolución práctica de problemas. De esta manera, y dada la relevancia y actualidad del trabajo en agentes/sistemas multi-agentes, el objetivo de la asignatura es aprender los conceptos básicos y las técnicas necesarias para poder desarrollar agentes y sistemas multi-agente inteligentes. Este objetivo se complementa con el estudio de las técnicas de inteligencia artificial que nos permitan dotar de “Inteligencia” a nuestros agentes artificiales.

Para ello, esta asignatura dispone de 3 créditos de teoría y 3 de prácticas, presentando el siguiente temario en su parte de teoría:

- Agentes Inteligentes:
 - Inteligencia Artificial Distribuida. Agentes y Tipos de Agentes. Conceptos. Modelos y Arquitecturas de Agentes. Aplicaciones. Prácticas en Laboratorio.
- Sistemas Multi-agente:
 - Arquitectura Multi-agente. Plataformas. Conceptos. Comunicación entre agentes. Coordinación en SMA. Métodos y Herramientas. Aplicaciones. Prácticas en Laboratorio.
- Incorporando “Inteligencia” en agentes:
 - Planificación: El problema de planificación. Conceptos. Planificación ordenada parcialmente. Grafos de planificación. Planificación y acción en el mundo real. Agentes basados en planificación. Aplicaciones. Prácticas en Laboratorio.
 - Razonamiento Basado en Casos: El ciclo CBR. Conceptos. Técnicas de razonamiento basado en casos. Representación de casos. Indexación. Almacenamiento y Recuperación. Adaptación. Herramientas. Tareas de clasificación y síntesis. Aplicaciones.

¹ <http://www.dsic.upv.es/users/ia/sma/tools/jgomas/>

- Teoría de la Decisión: Fundamentos de la teoría de la utilidad. Funciones de utilidad y de utilidad multi-atributo. Redes de Decisión. Problemas de decisión secuenciales. Iteración de Valores. Iteración de Políticas. Agentes basados en la teoría de la decisión.

El resto del artículo se estructura de la siguiente manera: la siguiente sección presenta los objetivos de las prácticas de la asignatura **Sistemas Inteligentes (SIN)** cuya modificación motiva este trabajo; a continuación se comentan las distintas herramientas que se han desarrollado para dicha modificación; posteriormente se describe la planificación de las sesiones de prácticas; después se describe el uso de la herramienta JGOMAS; por último se presentan unas conclusiones y comentarios sobre los resultados obtenidos.

2. Objetivos de las prácticas de SIN

El objetivo planteado en las prácticas de esta asignatura es el complementar, a un nivel práctico, los conceptos teóricos que sobre agentes y SMA se han visto en las clases de teoría. Para ello, se persigue que los alumnos empleen herramientas actuales. En concreto, para el caso de SMA, se buscó una plataforma

de SMA que fuese lo más actual posible, recayendo así la elección en la plataforma JADE. Esta plataforma es, sin lugar a dudas, una de las más utilizadas a nivel mundial en investigación y desarrollo.

Por tanto, los objetivos perseguidos al cambiar el contenido de las prácticas eran de diversa índole. Por una parte, mantener actualizadas dichas prácticas teniendo en cuenta las últimas herramientas desarrolladas, incluso aquellas que se encuentran en fase experimental. Por otra parte, el nuevo contenido de las prácticas debería de tener en cuenta que las herramientas incorporadas no tengan una complejidad de uso que provocase el rechazo de los alumnos. Por el contrario, éstas deberían motivar y captar el interés de los alumnos, potenciando su iniciativa para llevar a cabo ampliaciones a las exigencias mínimas de las prácticas.

3. Nuevas Herramientas para la Enseñanza de SIN

Como ya se ha mencionado con anterioridad, las motivaciones para crear un entorno diferente en el que realizar las prácticas de IA son muy diversas. Por ejemplo, se pretendía hacer más atractivas las técnicas de inteligencia artificial a

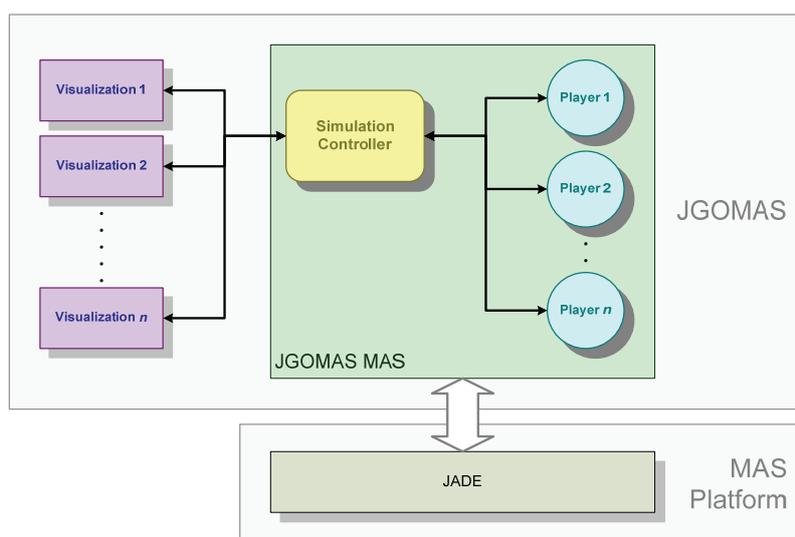


Figura 1. Composición de la herramienta JGOMAS.

los alumnos, así como incrementar la interacción entre los alumnos y el profesorado. Además, se quería facilitar en lo posible que los alumnos pudieran realizar o ampliar sus prácticas desde cualquier lugar, y no sólo desde los laboratorios docentes. Esto motivó tanto la creación de una nueva herramienta a usar por los alumnos para desarrollar las prácticas, como puesta en marcha de una Web dedicada a esta herramienta que permitiese la difusión de documentación, nuevas versiones, ejemplos y cualquier noticia en general de interés al alumno sobre la misma.

3.1. JGOMAS: Game Oriented Multi Agent System, based on JADE

Descripción del juego

JGOMAS consiste en un entorno para desarrollar y ejecutar agentes sobre mundos simulados en 3D. Concretamente, este entorno permite ejecutar agentes que pertenecen a dos equipos diferentes (rojo y azul, aliados y eje). Los equipos compiten en un juego del tipo “Capturar la bandera”, donde deben pugnar para capturar la bandera del oponente. Esta modalidad de juego se ha convertido en un estándar incluido en casi todos los juegos multi-jugador aparecidos desde el famoso videojuego **Quake** [1].

Por tanto, puesto que cada soldado puede verse como un agente, es muy fácil e intuitivo aplicar los sistemas multi-agente a este tipo de juegos, permitiendo la cooperación entre agentes del mismo equipo y la competición con los agentes del equipo contrario, resultando un entorno altamente competitivo para los diferentes agentes implicados.

De hecho, no es extraño encontrar aplicaciones de la tecnología de agentes al ámbito de los juegos, en general (como, por ejemplo, el juego de tablero desarrollado por S. Offerman et al. [2]), y de capturar la bandera, en particular. En este último caso, se pueden encontrar aplicaciones ad-hoc como el proyecto CTF [3], o aplicaciones a juegos comerciales [4] tales como el ya mencionado **Quake**.

Descripción de la herramienta desarrollada

Tal y como se ha mencionado anteriormente, JGOMAS es una herramienta que permite ejecutar un sistema multi-agente en un entorno

virtual 3D. JGOMAS está compuesto, principalmente, por dos subsistemas (Figura 1):

- Un Sistema Multi-Agente (JGOMAS MAS) ejecutándose sobre la plataforma de desarrollo y ejecución JADE. Dentro del SMA, existen dos tipos diferentes de agentes ejecutándose. Del primero de ellos existe un único agente controlador de la lógica del juego en curso, que además sirve de interfaz con los visualizadores del juego. Del segundo tipo existe un agente por cada uno de los jugadores de cada uno de los equipos. Estos agentes jugadores pueden desempeñar tres roles diferentes: soldado, médico u operador de campo.
- Un motor gráfico o visor de mundos virtuales 3D multiplataforma (implementado en C++ y OpenGL), que permite a los usuarios visualizar los agentes JGOMAS lanzados sobre JADE, viendo el transcurso de su ejecución dentro de un entorno virtual 3D. Así, es posible seguir visualmente como se comportan los componentes de cada equipo, y observar cuál de los equipos gana cada partida. En concreto, cada visualizador permite observar, en una única ventana, el entorno virtual en el que los agentes están inmersos. Además, es posible tener diversos visualizadores para satisfacer las necesidades de visualización de los diferentes usuarios.

3.2. Web

La idea de crear una Web específica para el entorno JGOMAS surgió a fin de poner a disposición de los alumnos un medio accesible desde cualquier lugar, donde en todo momento pudieran obtener la última versión del código disponible, contactar con los profesores o consultar los manuales de referencia.

Actualmente, desde la Web de JGOMAS es posible acceder a una breve explicación sobre qué consiste este entorno. Asimismo se explica cómo funciona JGOMAS y sobre qué componentes está basado. Desde la sección dedicada a la documentación, los usuarios pueden acceder a manuales de referencia en línea, así como descargar su versión en PDF si así lo prefieren. Desde esta sección también es

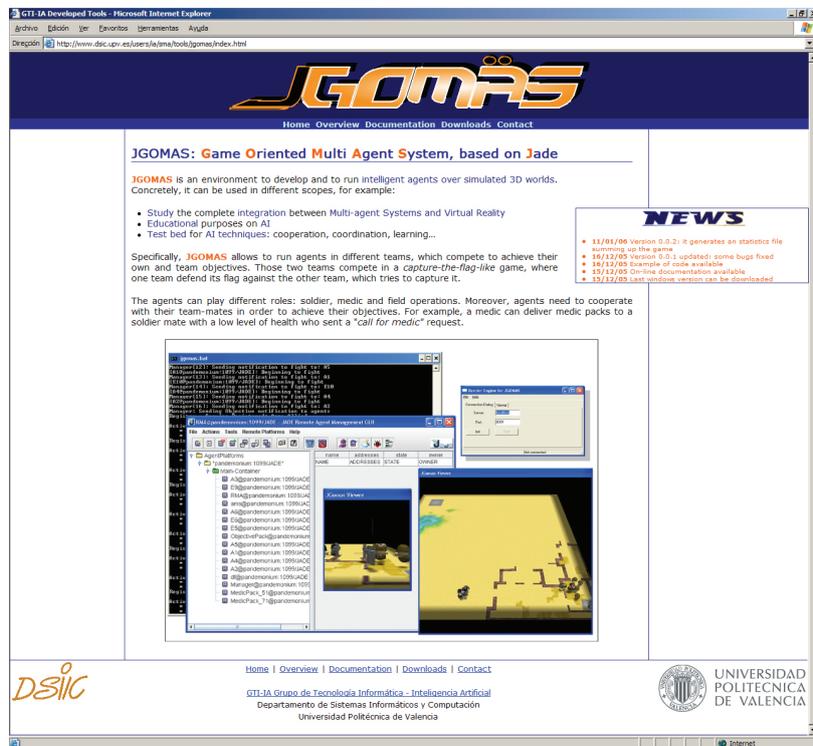


Figura 2. Página principal de la Web dedicada a la herramienta JGOMAS.

accesible todo el material didáctico de las prácticas, como por ejemplo las transparencias usadas en clase. De igual manera, está disponible una sección de descargas desde donde pueden obtener las últimas versiones del paquete JGOMAS, así como una breve explicación de cómo poner en marcha el entorno.

Por otra parte, todas las noticias relevantes sobre el entorno JGOMAS son accesibles desde la sección “NEWS” de la página principal de la Web (Figura 2), de forma que es sencillo hacer llegar a los alumnos que está disponible una nueva versión del entorno, etc. Asimismo, los alumnos pueden contactar a través de la Web con los profesores y desarrolladores de la herramienta, de forma que no sólo pueden consultar dudas, sino que también pueden proponer mejoras al entorno o la adición de características que les faciliten poner en práctica sus mejoras en el comportamiento básico de los agentes ofrecidos por JGOMAS.

4. Planificación de las Sesiones de Prácticas

Tal y como se ha comentado en la introducción de este artículo, hay 3 créditos de prácticas de la asignatura SIN. Estos créditos se traducen en sesiones semanales de 2 horas en el laboratorio. Estas sesiones se pueden clasificar de la siguiente manera:

1. Gestión de un agente para el control de un robot Khepera: Durante cuatro sesiones se presenta un agente diseñado para el control de un robot Khepera y se solicita al alumno que lo modifiquen para que realice un comportamiento determinado.
2. Gestión de un SMA: Esta parte de las prácticas es la que recoge las aportaciones objeto de este artículo. Se compone del resto

de sesiones de prácticas, dividiéndose a su vez en las siguientes partes:

2.1 Introducción a la plataforma JADE:

Durante 2 sesiones se presenta esta plataforma, tanto a nivel organizativo como a nivel de la biblioteca de funciones JAVA que incluye para poder crear agentes que funcionen en ella. Para ello, se utilizan tanto ejemplos como ejercicios que deben resolver los alumnos en estas sesiones.

2.2 Introducción a la herramienta JGOMAS:

Durante 2 sesiones se presenta la herramienta JGOMAS, desarrollada sobre JADE, junto con la biblioteca de funciones que complementan la biblioteca de JADE y que permite la creación de agentes de los tipos usados en JGOMAS. Durante estas sesiones se presentan también ejemplos y ejercicios que deben resolverse durante las mismas.

2.3 Creación de un equipo competitivo para JGOMAS:

El objetivo de las sesiones restantes de prácticas es que los alumnos hagan uso de la biblioteca de funciones de JGOMAS y lleven a la práctica los conocimientos adquiridos para construir un equipo de 10 agentes, que sea lo más eficiente posible, teniendo que cubrir como objetivos mínimos el que sea mejor que los equipos suministrados en la distribución de la herramienta. Es decir, que al enfrentarlos al equipo básico que se les entrega, consigan capturar la bandera y llevarla a su base en el caso en el que ataquen, y que sean capaces de impedir que el equipo contrario lo haga, en el caso en el que defiendan.

Al finalizar el cuatrimestre, se realiza una sesión de evaluación, donde se prueban los equipos programados por los alumnos contra el equipo básico que se distribuye en distintos escenarios. Además, se realiza una competición enfrentando los equipos de los alumnos entre ellos. La calificación obtenida por los alumnos dependerá en parte de su clasificación en la competición descrita, exigiendo siempre para poder aprobar que su equipo supere al equipo básico. Cabe destacar que los alumnos no disponen, previo a la sesión de evaluación, de los escenarios en los que

debe competir su equipo. Además, cada equipo se prueba en los escenarios actuando como atacante y como defensor.

5. Uso de la herramienta JGOMAS

El uso de esta herramienta supone la extensión o mejora del comportamiento de los agentes jugadores de un equipo y su posterior prueba en distintas condiciones (distintos escenarios y/o equipos a los que enfrentarse).

5.1. Ejecutar JGOMAS

Se puede encontrar la última versión de JGOMAS en la sección de descargas de su Web. Esta distribución incluye la plataforma, el visor de mundos virtuales 3D, mapas, documentación y un ejemplo listo para usar.

A la hora de ejecutar una partida con JGOMAS, ya que los agentes JGOMAS son agentes JADE, se pueden ejecutar directamente desde la GUI de JADE. Esta opción, sin embargo, no es recomendable debido a los parámetros que se deben pasar a los agentes al lanzarlos. Para facilitar el arranque, en lugar de utilizar la GUI de JADE se usarán una serie de *scripts* que se facilitan con la distribución de JGOMAS. En la Figura 3 se puede ver un ejemplo de ejecución de JGOMAS donde aparece la GUI de JADE, una consola de texto, y dos instancias del visor.

Los parámetros que necesita un agente en su lanzamiento dependen del tipo de agente. Así, un agente soldado necesita tan sólo un parámetro que indica el equipo al que pertenece (ALLIED o AXIS), mientras que el agente controlador de la lógica del juego espera los siguientes parámetros ordenados:

1. Número de agentes soldado que comenzarán la partida.
2. Mapa donde se jugará la partida.
3. Frecuencia de refresco de los visores (en milisegundos).
4. Duración máxima de la partida (en minutos).

Un ejemplo del lanzamiento de los agentes de una partida de JGOMAS es el siguiente:

```
java -classpath lib\jade.jar; lib\jadeTools.jar; lib\http.jar; lib\iiop.jar; .\JGomas.jar;. jade.Boot
```

-gui Manager:Classes.CManager(4 map_04 125 10)

A1:Classes.CMedic(ALLIED)

A2:Classes.CMedic(ALLIED)

E1:Classes.CMedic(AXIS)

E2:Classes.CMedic(AXIS)

Según el ejemplo anterior, el agente controlador (Manager Agent) comienza una partida con 4 jugadores en el mapa map_04 durante 10 minutos a 8 frames por segundo (125 milisegundos).

Además de los agentes que componen el MAS, se pueden lanzar uno o varios visores para poder ver la evolución del juego (encontrándose esta aplicación en la carpeta *GFX* de la distribución). Al lanzar el visor hay que indicarle la máquina y puerto en el que se encuentra el agente controlador, por medio del diálogo que aparece en la Figura 4 a). Además, una vez establecida la conexión, el visor no sólo permite ver la representación 3D del escenario y la interacción de los agentes, sino también ver el flujo de mensajes que va recibiendo, a través del diálogo que aparece en la pestaña *viewer* de esta primera ventana de conexión y que puede observarse en la Figura 4 b).

5.2. Evaluación de resultados obtenidos en una partida

Un aspecto que también resulta importante es la evaluación del resultado del juego. Un usuario va a ser capaz de evaluar lo que ha ocurrido durante el juego de dos formas, cuantitativa y cualitativamente.

Por una parte, un usuario podrá realizar una evaluación cualitativa gracias a la interfaz gráfica que permite ver la evolución del juego. Así, por ejemplo, un usuario puede comprobar si un agente se mueve correctamente con su nuevo algoritmo de generación de caminos, o si la distribución estratégica de los agentes en el mapa es la que él diseñó.

Por otro lado, la evaluación cuantitativa se facilita por medio de las estadísticas generadas al final del juego, permitiéndole al usuario comprobar la eficiencia de los agentes. De esta manera, por ejemplo se ofrecen valores como el número de paquetes de medicina repartidos frente a los que hayan sido recogidos por compañeros de equipo o por enemigos. Esto le permite al usuario tener un resultado numérico para compararlo con otros juegos realizados.

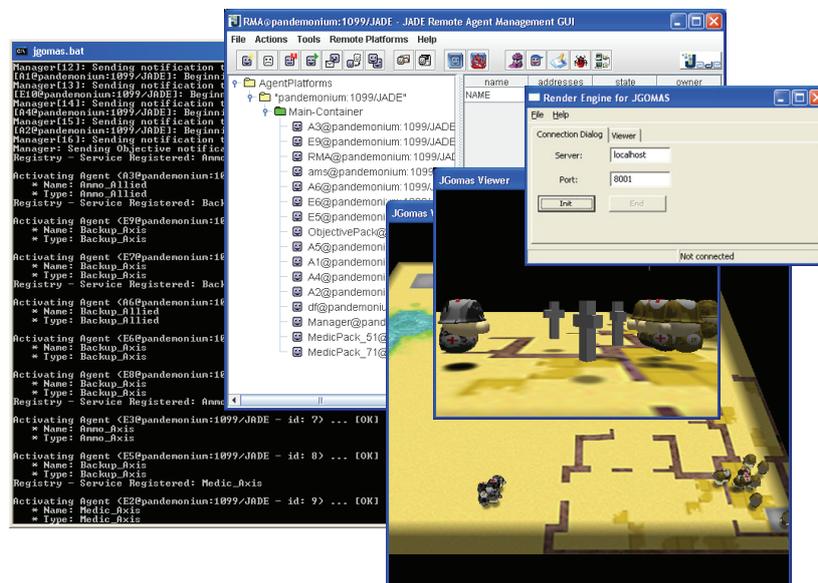


Figura 3. Ejemplo de ejecución de JGOMAS.

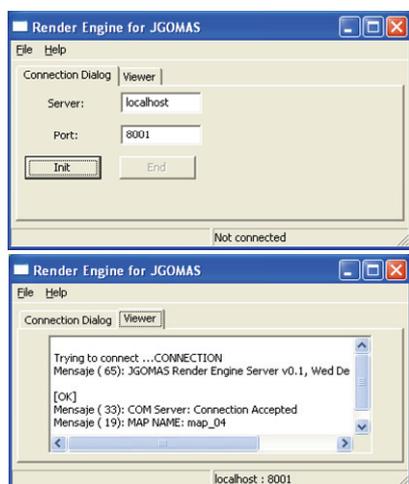


Figura 4. a) Diálogo de Conexión del visor.
b) Diálogo visualizador de mensajes del visor

5.3. Dotar de *inteligencia* a los jugadores

El SMA presentado está implementado mediante una serie de clases JAVA, que proveen a través de una sencilla biblioteca (jgomas.jar) una serie de servicios básicos. Los usuarios pueden extender estas clases, dando lugar a nuevas clases, a fin de obtener comportamientos más inteligentes para sus agentes, dando lugar a equipos de agentes más competitivos y efectivos. De esta manera, la taxonomía de agentes que proporciona JGOMAS viene determinada por las siguientes clases (Figura 5):

- *JGomas.Agent*: Ésta es la clase base de todos los agentes JGOMAS. De esta clase derivan el resto de clases de JGOMAS las correspondientes a los agentes *Manager*, *Pack* y *Troop*.
- *Manager*: Esta clase no es accesible por el usuario, pues corresponde al agente encargado de la coordinación y sincronización del resto de los agentes, así como de proveerles de todas la información relativa a su campo de visión.
- *Pack*: Representa a objetos relevantes del entorno (la bandera, los paquetes de salud y los de munición) y no son modificables por el usuario.

- *Troop*: Es la única clase de la taxonomía extensible por el usuario. Así, todos los agentes jugadores serán instancias de cualquiera de sus subclases: *Soldier*, *Medic* y *FieldOps*.
- *Soldier*: Corresponde a los agentes que llevarán el peso de la batalla, ofreciendo apoyo a su equipo.
- *Medic*: Esta clase de agentes ofrece un servicio médico, de forma que cualquier agente de su equipo que tenga un nivel de salud bajo puede pedirles ayuda. En esta situación un agente *Medic*, crearía un paquete de salud y lo dejaría al alcance del solicitante.
- *FieldOps*: Corresponde a los agentes encargados de suministrar munición al equipo. Así, creará paquetes de munición frente a las demandas de sus compañeros de equipo.

Por lo tanto, para poder dotar de inteligencia a un equipo de agentes JGOMAS, el usuario ha de crear una nueva clase de agente y modificar algunos métodos sobrecargables, así como añadir métodos nuevos. En la ya mencionada Web de JGOMAS se pueden encontrar diversos manuales y ejemplos² que pueden ayudar a conseguirlo.

6. Resultados observados y conclusiones

Se han desarrollado y puesto en marcha una nueva herramienta así como una Web ad-hoc para su uso dentro del marco de las prácticas docentes de Inteligencia Artificial. En concreto, la herramienta desarrollada, JGOMAS, es multi-plataforma y sigue los estándares fijados dentro de los SMAs, como por ejemplo FIPA. Así, esta aplicación puede también ser usada como simulador para la coordinación, comunicación y algoritmos de aprendizaje dentro del campo de los MAS, o de la IA en general. Para conseguir esto, JGOMAS permite al usuario añadir sus propias modificaciones de código. Además, JGOMAS puede ser usado para estudiar la integración de los MAS y de los sistemas de realidad virtual.

Por otro lado, la puesta en marcha de las nuevas sesiones de prácticas ha resultado muy

²Manual de usuario, manual de referencia y "How to use JGOMAS API: programming examples", en: <http://www.dsic.upv.es/users/ia/sma/tools/jgomas/archivos/documentation/>

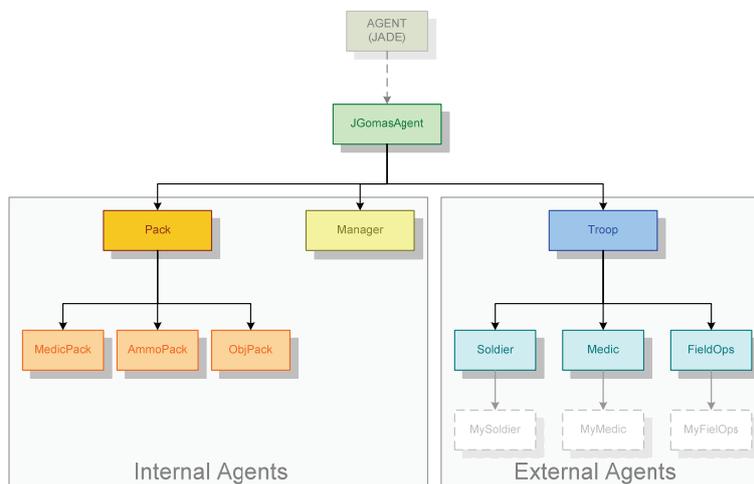


Figura 5. Taxonomía de agentes de JGOMAS.

satisfactoria, tanto por la acogida de los alumnos como por los resultados obtenidos en la evaluación de las mismas.

Así, el atractivo del tema de las prácticas junto con la competición establecida en la sesión de evaluación ha motivado a los alumnos a llevar a la práctica muchas de las técnicas explicadas en teoría, así como desarrollar estrategias innovadoras para sus equipos, con el objetivo de obtener el equipo más “inteligente”.

En concreto, a esta sesión de evaluación se presentaron el 70% de los alumnos matriculados en la asignatura, formándose un total de 8 equipos, de los cuales tan sólo uno no fue capaz de superar al equipo básico suministrado. Las estrategias de los equipos finalistas fueron:

- Campeón: Utilizaban una estrategia similar tanto en defensa como en ataque. Ésta se basaba en un algoritmo A* y una buena coordinación de sus tropas, para ir al encuentro de las tropas contrarias a fin de eliminarlos.
- Subcampeón: En defensa, dividían a sus soldados en dos grupos que hacían guardia alrededor de la bandera en trayectoria circular y en sentidos contrarios. Así minimizaban el tiempo que tardaban en detectar un enemigo. Por el contrario, en ataque, todos los agentes del equipo son médicos *pacifistas*. Su estrategia básica era la velocidad (no

perdiendo el tiempo en apuntar ni disparar). Además, eran capaces de suministrarse paquetes de salud.

Agradecimientos

Este trabajo ha sido parcialmente financiado mediante los proyectos CICYT TIC2003-07369-C02-01 y TIN2005-03395, en parte sufragados con fondos FEDER.

Referencias

- [1] id Software, Inc. Comprobado el 31-1-2006. <http://www.idsoftware.com/games/quake/quake4/>
- [2] Offermann, S., Ortmann, J., Reese, C.: Agent Based settler game. Part of NETDEMO, demonstration at international conference on Autonomous Agents and Multi Agent Systems, AAMAS-2005 (2005).
- [3] The Experimental Knowledge Systems Laboratory. Department of Computer Science at the University of Massachusetts Amherst. <http://www-eksl.cs.umass.edu/>
- [4] van Lent, M., Laird, J., Buckman, J., Hartford, J., Houchard, S., Steinkraus, K., Tedrake, R.: Intelligent Agents in Computer Games. American Association for Artificial Intelligence (1999).