

# Una propuesta de definición para Plataformas de Desarrollo y Plataformas de Ejecución en Sistemas Multi-agente

Tulio J. Marchetti  
tjm@cs.uns.edu.ar

Alejandro J. García  
ajg@cs.uns.edu.ar

Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA)  
Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur  
Av. Alem 1253, (8000) Bahía Blanca, Argentina  
Tel: (0291) 459-5135 / Fax: (0291) 459-5136

**Palabras clave:** Sistemas Multi-agente. Inteligencia Artificial Distribuida. Plataformas de desarrollo y ejecución

El objetivo central de este trabajo es proponer definiciones para los conceptos de *plataforma de desarrollo* y *plataforma de ejecución* de sistemas multi-agente. El motivo principal es que ambos conceptos son usualmente presentados en la literatura utilizando únicamente su significado intuitivo. Lamentablemente es difícil encontrar una definición concreta de estos términos y aún no existe un verdadero consenso sobre su significado y alcance.

Otro de los objetivos del trabajo es contrastar las definiciones propuestas, con las plataformas JACK y ZEUS. También se analiza la posibilidad de utilizar estas plataformas para desarrollar agentes LEGO [1], y se presenta una propuesta propia de plataforma para el desarrollo de estos agentes.

## Introducción

De la misma manera que en la literatura no existe aún un acuerdo definitivo sobre la definición de términos como *inteligencia* o *agente*, algo similar ocurre con el término *plataforma*. En la mayoría de los trabajos donde se describen plataformas de desarrollo o plataformas de ejecución de sistemas multi-agente no hay una definición concreta de estos términos. Esto dificulta el análisis de una plataforma debido a que no existe una definición con la cual se pueda comparar. A continuación, se presentan dos caracterizaciones del término plataforma de agentes que se han encontrado en la literatura.

Por un lado, en [9], se plantea la siguiente aproximación a la definición de *plataforma de agentes*: “un lugar donde los agentes viven, que provee la administración de éstos (creación, terminación, seguridad), que brinda servicios de comunicación entre agentes y que provee servicios de directorio de agentes”.

Sin embargo, lo que presenta la definición anterior son sólo las características que debe poseer una plataforma de agentes y no una plataforma de desarrollo o de ejecución. Esto se debe a que omite características propias de las plataformas de desarrollo, al igual que omite otras características propias de las plataformas de ejecución de sistemas multi-agente.

Por otro lado, en la opinión de Fabio Bellifemine [2], quien forma parte del grupo de desarrollo de JADE, una plataforma de agentes debe:

---

Financiado parcialmente por SeCyT Universidad Nacional del Sur (Subsidio: 24/ZN09) y por la Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 Nro 13096)

- Proveer servicios a los agentes y un entorno de vida seguro. Estos servicios pueden también estar implementados con agentes.
- Una herramienta de desarrollo no debería ser parte de la plataforma.
- Los servicios más importantes, en su opinión, son el transporte de mensajes, protocolos de interacción y ontologías.

En [2], al igual que en [9], no existe una definición que caracterice a una plataforma de desarrollo o una plataforma de ejecución de sistemas multi-agente. En ambos trabajos se expresan algunas características que debe cumplir una plataforma, pero a nuestro entender se omiten algunos elementos importantes. Es por este motivo, que en este trabajo presentaremos una propuesta de definición para los caracterizar a una plataforma de desarrollo de sistemas multi-agente y una plataforma de ejecución de sistemas multi-agente.

## Definiciones propuestas

En esta sección presentaremos las definiciones propuestas para plataforma de desarrollo y plataforma de ejecución de sistemas multi-agente.

**Definición 1 (Plataforma de desarrollo de sistemas multi-agente)** *Una plataforma de desarrollo de sistemas multi-agente es una herramienta que permite el diseño y desarrollo de un sistema formado por un grupo de agentes de software, los cuales tendrán una arquitectura básica determinada, y podrán interactuar entre ellos para resolver los problemas para los cuales el sistema ha sido diseñado. Esta herramienta debe poseer la capacidad de generar automáticamente el código de los agentes, el cuál proveerá su funcionalidad básica y las primitivas necesarias para la interacción con otros agentes del sistema. Además, esta herramienta debe proveer la posibilidad de agregar, en forma manual, código personalizado a los agentes en algún lenguaje de programación, con el fin de dotar al agente con mayor funcionalidad, inteligencia, etc. Para esto, la plataforma debería poseer un compilador para generar el código final de cada agente, incluyendo su código personalizado.*

Siguiendo la definición, anterior sería deseable que la plataforma de desarrollo de sistemas multi-agente, además cumpliera con las siguientes características:

- Permitir el desarrollo de un sistema en forma rápida, sencilla, lógica y comprensible.
- Debe ser una aplicación “liviana” en la ejecución.
- Debe poseer una interface gráfica, lo que ayuda para lograr que sea sencilla y comprensible.
- La arquitectura de los agentes y las primitivas de interacción deberían respetar algún estándar internacional.

Una plataforma de ejecución de agentes debe proveer los servicios para garantizar la correcta ejecución de un grupo de agentes, los cuales pueden haber sido construidos con una plataforma de desarrollo.

**Definición 2 (Plataforma de ejecución para sistemas multi-agente)** *Una plataforma de ejecución de agentes debe proveer los servicios necesarios para la ejecución de un conjunto de agentes en un entorno distribuido y posiblemente heterogéneo. Debe contar con los servicios de transporte de mensajes, seguridad y directorio de agentes.*

## Algunas plataformas

En trabajos anteriores tales como [7, 8] analizamos otras plataformas de desarrollo de agentes. Entre ellas podemos mencionar brevemente algunas, así como también tratar de clasificarlas de acuerdo a la definición de plataforma antes mencionada.

### JACK

JACK [3, 5] provee un entorno de desarrollo orientado a agentes construido sobre Java y completamente integrado con este lenguaje de programación. Incluye todas las componentes del entorno de desarrollo de Java, y además incluye las siguientes extensiones específicas para implementar el comportamiento de los agentes:

- Define nuevas clases base, interfaces y métodos
- Provee extensiones a la sintaxis de Java para soportar clases orientadas a agentes
- Provee extensiones semánticas para soportar la ejecución del modelo

La plataforma JACK es una plataforma de desarrollo y ejecución de agentes. Cuenta con una interfase gráfica amigable que hace que su utilización sea sencilla, es bastante “pesada” en ejecución, provee todo lo necesario para la ejecución de los agentes, permite generar el código de los agentes y además personalizar el código de los mismos. Posee un analizador para chequear que el código agregado a los agente o a archivos externos a los agentes sea correcto. Como última característica posee un entorno de ejecución que permite visualizar a los agentes desarrollados fácilmente.

### ZEUS

ZEUS [4, 10] es una herramienta para construir aplicaciones multi-agente colaborativas. que provee un entorno integrado para el desarrollo rápido de sistemas. ZEUS define una metodología de diseño de sistemas multi-agente y lo soporta mediante un entorno visual para capturar las especificaciones de los agentes. Estas especificaciones son luego utilizadas para generar el código fuente en Java.

Con ZEUS tenemos el caso de una plataforma de desarrollo de agente. Cumple con todas las características de una plataforma de desarrollo pero no es una plataforma de ejecución. Cuenta con la interfase gráfica, es “liviana” en ejecución, permite generar el código de los agentes y provee todos los servicios para la ejecución de los agentes. A pesar de proveer todo esto, no provee un entorno de ejecución dentro de la plataforma, la ejecución es externa a la misma.

## Propuesta de aplicación

Viendo las características mas importantes que poseen las plataformas de desarrollo de sistemas multi-agente, se intenta bosquejar el diseño de una para el desarrollo de agentes LEGO.

LEGO MINDSTORM [6] es un kit de ensamblaje para la construcción de robots mediante el uso de ladrillos LEGO [1], su nombre completo es LEGO MINDSTORM ROBOTICS INVENTION SYSTEM, pero en adelante será referenciado como RIS. El RIS, además de los comunes ladrillos LEGO

de diversas formas, tamaños y colores, encontramos motores, sensores, transmisiones, ejes, piñones y lo mas importante una computadora llamada Robotics Command Explorer que será referenciada como RCX.

El RIS permite la construcción de robots que mediante el uso de esos sensores, motores y del RCX pueden: obtener información del mundo que los rodea, movilizarse por un entorno y por medio del RCX comunicar por via infrarroja los datos que percibe a una computadora. Así mismo, la programación del RCX se realiza en NQC (Not Quite C), pero en la interacción con una computadora, el programa que en ella se encuentra, puede realizarse en varios lenguajes.

El código NQC que se encuentra en el RCX provee las primitivas básicas para el movimiento, sensado, envío y recepción de mensajes. El agente que se encuentra en la computadora (el agente LEGO), es el encargado de analizar los datos de entrada de los sensores, tomar decisiones y enviar acciones al RCX para que este las ejecute.

Las características mas destacadas de la plataforma pueden resumirse en la siguiente lista, la cual muestra a la vez las que formarán parte de la propuesta:

- Interface gráfica
- Editor de código (Prolog)
- Compilador (AMZI Prolog)
- Generación de código (AMZI Prolog y Delphi)

El objetivo de poseer esta plataforma es agrupar todo el trabajo en una sola aplicación. Cuando se genera el agente, la plataforma esta generando una aplicación Delphi estándar que posee todas las primitivas de comunicación y manejo del robot a través del puerto, junto con el agente compilado y linkeado. Una vez que se ha generado el agente, la compilación del código Prolog y la interface Delphi, el agente esta listo para ser ejecutado. Para la ejecución sólo se debe utilizar el archivo Delphi *.exe* generado por la plataforma. Esto se encargará de cargar la definición de los predicados de comunicación y también de la carga del agente Prolog.

Hay que aclarar que la plataforma propuesta es solamente una plataforma de desarrollo de agentes. Cumple con todas las características: es una aplicación liviana, posee una interfase gráfica, permite la generación de código y la personalización del código de los agentes pero no posee un entorno de ejecución de agentes.

Para el desarrollo de una plataforma, para la creación de agentes LEGO se intentó utilizar alguna de las plataformas vistas anteriormente y analizadas en [7, 8]. Como el primer punto a tener en cuenta es que ninguna de estas plataformas permite la creación de agentes en Prolog ya que el código final de todas ellas es Java. Poder generar a los agentes en código Prolog fue desde un primer momento un objetivo para el desarrollo de este tipo de agentes, por lo que ya tenemos un punto muy importante para decir que ninguna de las plataformas nos sería de utilidad para la implementación de este tipo de agentes.

Otro punto a tener presente es el hecho que, aún renunciando a desarrollar los agentes en Prolog y desarrollarlos en Java, la experiencia de desarrollo en ese lenguaje no tuvo los resultados deseados. Cuando se intento hacer uso de ese lenguaje para el desarrollo de este tipo de agentes, nos encontramos con retardos muy grandes en la utilización de la torre infrarroja, con lo que la comunicación y la interacción robot-PC se veía muy demorada.

## Conclusiones y trabajo futuro

En este trabajo fueron propuestas definiciones para los conceptos de *plataforma de desarrollo* y *plataforma de ejecución* de sistemas multi-agente. El motivo principal es que ambos conceptos son usualmente presentados en la literatura utilizando únicamente su significado intuitivo. También se contrastaron las definiciones propuestas, con las plataformas JACK y ZEUS, y se analizó el desarrollo de una plataforma para agentes LEGO.

Como trabajo futuro se planea contrastar las definiciones propuestas con otras plataformas existentes buscando encontrar una definición definitiva a estos términos. Por otro lado como línea de investigación se planea realizar un diseño de una plataforma de desarrollo y ejecución de agentes de propósito general.

## Referencias

- [1] LEGO. <http://www.lego.com/>, 2003.
- [2] F. Bellifemine. *Opinión personal de plataforma de agentes*. <http://sharon.cselt.it/projects/jade/jade-develop-archive-Feb2002/0655.html>, 2003.
- [3] Hodgson Busetta, Rönnquist and Lucas. Jack intelligent agents - components for intelligent agents in java. Agent Oriented Software Pty. Ltd., 1999.
- [4] Collins and Ndumu. *The Zeus Agent Building Toolkit*. ZEUS Technical Manual, 1999.
- [5] JACK. *JACK Intelligent Agents*. Agent Oriented Software Pty. Ltd., <http://www.agent-software.com/>, 2003.
- [6] LEGO. *LEGO MINDSTORM Robotics Invention System*. <http://www.legomindstorm.com/>, 2003.
- [7] Tulio Marchetti and Alejandro J. García. Evaluación de plataformas para el desarrollo de sistemas multiagente. *IX Congreso Argentino de Ciencias de la Computación. La Plata, Argentina*, pages 625–636, 2003.
- [8] Tulio Marchetti and Alejandro J. García. Plataformas para desarrollo de sistemas multiagente. un análisis comparativo. *V Workshop de Investigadores en Ciencias de la Computación. Tandil, Argentina*, pages 389–393, 2003.
- [9] Moscow Institute of Physics and Technology, <http://zadig.fizteh.ru/agents/ic03-dotNAP.ppt>. *Innovation through imagination*, 2003.
- [10] ZEUS. *The ZEUS Agent Building Tool*. British Telecommunications, <http://more.btexact.com/projects/agents/zeus/>, 2002.