

# Desarrollo de Sistemas Multi-Agentes

**Coordinado por: Analía Amandi**

ISISTAN Research Institute, Facultad de Ciencias Exactas,  
Universidad Nacional del Centro de la Pcia. de Buenos Aires  
Campus Paraje Arroyo Seco - (B7001BBO) Tandil - Bs. As., Argentina  
email: amandi@exa.unicen.edu.ar

## 1. Introducción

El desarrollo de sistemas multi-agentes es hoy en día uno de los temas de mayor discusión en el área de Agentes. Tanto la aplicación de resultados del área de Ingeniería de Software como la necesidad de generación de nuevos métodos y herramientas específicas para el desarrollo de agentes son temas que se intentan esclarecer.

Estos objetivos son seguidos hoy cuando ya numerosos agentes y sistemas multi-agentes han sido desarrollados, y a pesar que se continúa discutiendo qué es un agente. En la discusión de qué es un agente, diferentes opiniones se pueden escuchar en relación a la necesidad de la inclusión de varias características.

Básicamente, un agente es una entidad de software que exhibe un comportamiento autónomo y un sistema multi-agente es un conjunto de agentes que tienen la capacidad de interactuar en un entorno común. Así, agentes en un entorno con otros agentes poseen capacidades como la comunicación, negociación, y coordinación. Características que podemos considerar opcionales son encontradas en varios tipos de agentes, como la movilidad y, la

necesidad de interacción con usuarios y el consiguiente aprendizaje de su comportamiento.

Las diferentes características que puede poseer un agente ha permitido la definición de diferentes tipos de agentes. Así, los agentes móviles son aquellos que pueden moverse en una red distribuyendo su ejecución. Los agentes de interfaz son aquellos que asisten a usuarios, para lo cual deben aprender de ellos a partir de instrucciones explícitas o de la observación de su comportamiento.

Así, los comúnmente denominados agentes o agentes inteligentes o agentes de software o agentes autónomos o softbots nos presentan problemas para su construcción relacionados a la toma de decisiones en forma autónoma, la coordinación, la negociación, el manejo de estados mentales y la comunicación. Adicionalmente a estos problemas, los agentes móviles y los de interfaz nos presentan otros inconvenientes que debemos tener en cuenta en el desarrollo de estos tipos de agentes, como la movilidad y el aprendizaje de preferencias de usuarios.

Estos problemas tienen que ser evaluados en el sentido de aplicar métodos y herramientas existentes o nuevas propuestas basadas en éstos, para así

aprovechar los resultados alcanzados por investigadores del área de ingeniería de software. De esta manera, aquellos que deseen desarrollar agentes simples o sistemas multi-agentes pueden facilitar tanto la construcción como el reuso de estos agentes.

En síntesis, estos problemas que están siendo atacados actualmente motivaron la realización de esta monografía. En esta edición, se presentan varios artículos que nos permiten ver el estado de las investigaciones en Iberoamérica en este campo que limita entre la Inteligencia Artificial y la Ingeniería de Software.

A continuación se presentan los artículos que muestran los avances en este campo por parte de investigadores iberoamericanos.

Los dos primeros artículos presentan muy interesantes desarrollos de sistemas multi-agentes. El primero de ellos en el dominio médico y fue construido utilizando un framework para el desarrollo de agentes Java denominado Jade. El segundo con relación a subastas electrónicas en la Web fue implementado utilizando los lenguajes Java y JSP.

A continuación, una herramienta que permite la construcción de sistemas multi-agentes a partir del reuso de componentes desarrollados dentro del paradigma de la orientación a objetos. Esta herramienta, denominada *Bubble*, es calificada como un proto-framework que codifica en Java los componentes y comportamientos comunes a entidades colaborativas, posibilitando a través de ellos la construcción de otros frameworks más específicos de agentes.

En cuarto lugar, conceptos teóricos acerca de la complejidad de las relaciones entre agentes en un

sistema multi-agente son detalladamente introducidos de una manera formal.

Luego, detalles relacionados a la interacción en términos de conflictos producidos entre varios agentes, desde el punto de vista de su desarrollo, son expuestos en el siguiente artículo. Este trabajo fue desarrollado utilizando el lenguaje orientado a objetos Java y un intérprete Prolog aumentado con un algoritmo de programación lógica inductiva denominado Progol.

Después de la presentación del trabajo en simuladores de conflictos, un lenguaje para la programación de agentes es presentado. Este lenguaje, denominado JavaLog, permite desarrollar agentes como objetos Java que tienen la posibilidad de manipular complejos estados mentales codificados en un lenguaje lógico como Prolog o extensiones de éste. Esta integración multi-paradigma entre Java y Prolog está basada en el manejo de módulos lógicos.

En el siguiente artículo, se sitúan los agentes en el Web, introduciendo esta problemática y sus características particulares, ejemplificando con el dominio de comercio electrónico.

Finalmente, una propuesta para potenciar servicios de servidores Web con características más inteligentes es presentada. Así, entidades denominadas *Brainlets* permiten que agentes puedan moverse entre diferentes sitios que soporten la ejecución de código JavaLog, habilitando así a agentes a distribuir el procesamiento de su información y sus decisiones en diferentes y distantes sitios Internet.

Como podemos observar, varios trabajos relevantes al desarrollo de agentes y sistemas multi-agentes son presentados en esta monografía. Los resultados

de estos trabajos nos habilitan a pensar que nuestra comunidad iberoamericana está mostrando buenos resultados en diferentes aspectos de varios tipos de agentes relacionados al desarrollo.

### **Agradecimientos**

Agradezco la colaboración de los miembros del Instituto ISISTAN por su colaboración, especialmente a Alejandro Zunino y a Andrés Díaz Pace.