

# Aportación del paradigma orientado a agentes en el desarrollo de tutores de *e-learning*

Andrés Castillo, Oscar Sanjuán, Luis Joyanes

Dpto. de Lenguajes e Ingeniería del Software  
Universidad Pontificia de Salamanca  
28040 Madrid

e-mail: {andres.castillo, oscar.sanjuán, luis.joyanes}@ups.es

## Resumen

Los sistemas multimedia son los que desde hace pocos años se responsabilizaron de la formación basada en sistemas electrónicos. Las actuales razones económicas con una acentuada crisis, cambios tecnológicos o la creciente necesidad de una formación continua hace que la responsabilidad recaiga sobre *E-learning*, es decir, cualquier método que permita distribuir contenido útil para el aprendizaje a través de Internet y en el que intervienen profesores, alumnos y el centro de formación. Además, Internet, el nacimiento de las intranets y la evolución hacia los portales de empleados han propiciado el acceso a plataformas de *e-learning*.

Entre los nuevos roles impuestos por el *e-learning* se encuentran los tutores o formadores, alumnos, empresas de formación y los gestores de contenidos. El tutor virtual deberá ser el dinamizador que, junto con los contenidos, haga que el sistema funcione. Tendrá como actividad principal informar, motivar y comprometerse con el sistema. Mientras a los alumnos les compete la navegación interactiva por los contenidos, actividades y servicios que activan el aprendizaje de los alumnos, el tutor debe comprometerse con su aprendizaje.

Este artículo describe la aportación del paradigma de orientación a agentes en el desarrollo de los tutores virtuales de *e-learning*. Para ello se describe una arquitectura concreta que se basa en agentes y utiliza otras tecnologías avanzadas de información.

## 1. Introducción

En la actualidad el *e-learning* está situado en entornos como la empresa para la formación de sus propios trabajadores, en la banca para la

formación continua, etc.; y realmente donde el campo de aplicación es más amplio es en el área de la educación, universidad, formación a distancia, institutos técnicos, educación de personas adultas, etc. Es en esta área donde se encuentra un amplio campo de estudio y aplicación de los mejores recursos. Al contrario que en el campo de la empresa en el que se dispone de un amplio abanico de recursos técnicos ya disponibles que se aplican en otros campos.

En los próximos años el crecimiento de la aplicación de *e-learning* tiene previsto un crecimiento con una estimación del veinte por ciento en el total de la formación en 2005. El 60% de las empresas españolas tendrán implantados sistemas *e-learning* en 2004, según un estudio de la Asociación Española de Formación Online (AEFOL). Por otro lado, expertos que han intervenido en el estudio han coincidido en señalar entre las barreras para el desarrollo del *e-learning* algunos "frenos de tipo tecnológico", como la velocidad, la seguridad de accesos o, en el ámbito interno de la empresa, las reticencias por parte de los departamentos de sistemas.

Entre las barreras de tipo pedagógico señala la falta de conocimiento sobre metodologías de aprendizaje en modalidad 'online' y sugiere la necesidad de centrarse en los contenidos y pedagogía. También apunta que existen problemas a la hora de adecuar el '*e-learning*' al aprendizaje efectivo de cada colectivo y cada competencia o la constatación de la dificultad para concentrarse ante la pantalla más de veinte minutos seguidos.

La decisión de adoptar una plataforma *e-learning* concreta vendrá determinada por una serie de indicadores a tener en cuenta: por un lado, la gestión de contenidos y el hecho de que no se sientan condicionados a ninguna plataforma; por otro, la integración de distintas fuentes y formatos y, por último, la tecnología. Tampoco habrá que descuidar el método de desarrollo, donde analizar

si se respetan los estándares *e-learning* y si su uso resulta fácil e intuitivo.

Este documento pretende mostrar una visión simplificada de la arquitectura tecnológica del sistema de *e-learning* propuesto haciendo especial hincapié en la aportación de cada uno de los componentes tecnológicos clave que conforman el sistema que se está desarrollando en la Facultad de Informática de la Universidad Pontificia de Salamanca.

## 2. Características principales del sistema

Para lograr los requisitos básicos de un sistema de enseñanza, se está desarrollando por parte del Grupo de Investigación de Agentes y Gestión del Conocimiento de la UPSAM un sistema basado en agentes inteligentes, dicho Sistema de Tutorización, tiene como objetivo generar el aprendizaje en el alumno. La inteligencia de los Agentes se identifica en dos vertientes, por una parte la capacidad de guiar o autorizar al estudiante, y por otro lado su capacidad de adaptación, esta característica se logra mediante técnicas de programación evolutiva.

El Sistema Mediador es capaz de almacenar elementos clasificables, independientes, comprensibles, útiles y reusables, para lograr este objetivo el material didáctico es gestionado como objetos de aprendizaje. Al tratarse de un sistema cognitivo, donde la captura, almacenamiento, recuperación y difusión son elementos fundamentales, es necesario utilizar técnicas y estrategias tomadas de la Ingeniería del Conocimiento.

Una de las mayores deficiencias de los sistemas actuales es su incapacidad para interactuar con otros sistemas heterogéneos, lo que los hace muy costosos y poco reutilizables. En este sentido, el mecanismo básico de comunicación entre aplicaciones está basado en el uso de Servicios *Web*. En un sistema de estas características, los procesos de carga de la base de conocimiento y los mecanismos para acceder a fuentes de información externa, de las que muchas veces es imposible conocer su estructura exacta u organización, constituyen la razón que obliga a pensar en el uso de técnicas de minería de datos, para la extracción de la información de *Dataware House* para la creación de un registro de gran utilidad para ulteriores mejoras en los servicios del sistema.

La interacción con el usuario juega un papel fundamental a la hora de promover el aprendizaje, con este fin se recurre a técnicas tomadas de la Ingeniería de Software y en particular de la ingeniería *web*, para lograr construir un sistema usable y accesible.

En lo referente al medio de acceso, para facilitar el que se pueda acceder al sistema en cualquier momento y desde cualquier lugar, el desarrollo cuenta con la cualidad de integrar múltiples medios de comunicación (inalámbricos y tradicionales), múltiples protocolos (HTTP, WAP), múltiples interfaces (*web* o para dispositivos móviles) e incluso sistemas de representación (auditiva, multimedia, basada en realidad virtual)

En cualquier proyecto de software es necesario seleccionar y adaptar un proceso metodológico que permita llevar a buen puerto el proyecto sin desviaciones significativas en lo referente a costes y tiempos. El desarrollo se realiza mediante una metodología iterativa e incremental híbrida, orientada a agentes y basada en el Proceso Unificado de Desarrollo de Software.

Cada uno de los elementos mencionados se comenta en las siguientes secciones.

## 3. Arquitectura tecnológica

El avance tecnológico en las comunicaciones lleva al planteamiento de nuevos escenarios en los que términos como ubicuidad, sistemas inteligentes, sistemas distribuidos, alta disponibilidad, redes seguras, etc. hacen necesario el uso de nuevas técnicas, nuevas metodologías, y nuevos sistemas.

En vista de estos avances que necesitan nuevos enfoques, campos tan diferentes a priori como pueden ser las redes y los sistemas distribuidos frente a la Inteligencia Artificial, deben trabajar juntos para encaminarse a lograr estos objetivos.

Los sistemas tradicionales de la Inteligencia Artificial se conciben individualmente como un agente monolítico cuyo comportamiento sería descrito como 'racional' por un observador externo. A principios de los años ochenta los sistemas para resolución distribuida de problemas se caracterizaban por una forma de actuación concurrente en los diferentes nodos de una red, en general con control centralizado, de forma que los diferentes componentes permanecen impasibles

ante las actuaciones del resto de los componentes de la red. En este caso el objeto de estudio es la coordinación para resolución distribuida de problemas.

Al principio de los años noventa aparecen sistemas multiagente con control descentralizado y con módulos reusables. Los agentes de un sistema multiagente se conciben como independientes de un problema en concreto y junto con ello se dota al sistema de protocolos de comunicación suficientemente genéricos.

Recientemente los trabajos se han orientado al estudio de la interoperabilidad de sistemas heterogéneos distribuidos o de agentes autónomos y a su forma de adaptarse a entornos dinámicos.

#### 4. Servicios *Web*

Por servicios web (*web services*) se entiende el uso de tecnología Internet (web, correo, ftp, etc.) para interconectar procesos, generalmente transaccionales, entre empresas y sus clientes o proveedores. Este tipo de procesos son del tipo extranet/intranet, ya que no están accesibles al público general. El tipo de estructuras de referencia, se ha venido desarrollando en diferentes sectores industriales.

En referencia a las aplicaciones *web*, su construcción puede resultar sumamente complicada, debido sobretodo a que la *web* está pensada para que los datos sean consultados de modo sencillo y atractivo, lo que viene a complicar el desarrollo de aplicaciones cuya finalidad es integrar información procedente de fuentes diversas. Hasta hace un tiempo, la programación distribuida se basaba únicamente en un sistema básico de llamadas a procedimientos remotos. Este sistema permite la comunicación de procedimientos desarrollados en diferentes equipos y lenguajes que pueden, sin embargo, interactuar entre sí, con total independencia de los equipos en que residen dentro de la red que los conecta. El problema principal del método, proviene de la dificultad y complejidad propia de las aplicaciones distribuidas complejas. Por otra parte, a menudo las aplicaciones construidas en diferentes plataformas, tienen dificultades a la hora de lograr la comunicación; o sea que la compatibilidad está lejos de ser total.

Se precisa utilizar un denominador común, para tratar con el enorme número de entidades

heterogéneas, que existen bajo Internet. Es decir, cuando grandes cantidades de información se transfieren de un servicio a un cliente, y viceversa, el proceso requiere el uso de algún estándar que todos soporten en Internet. El estándar más común es HTTP, utilizado en la mayoría de los navegadores, para acceder a páginas *web*.

En la actualidad, el estándar más usado en la transmisión de datos a través de HTTP, es XML. Microsoft ha desarrollado el servicio *web*, basándose en el protocolo HTTP y el estándar XML, de modo que los servidores aceptarán peticiones de objetos cliente, usando HTTP y XML. Escribir un servicio *web* es, en realidad, tan fácil como escribir una aplicación servidora, tal como si esta fuese a tratar peticiones de clientes locales. Simplemente, se utiliza un atributo para indicar que se trata de un servicio *web*, y Microsoft.NET hace el resto, lo que equivale a decir que creará la infraestructura que acepte peticiones llegadas a través de HTTP, y las transformará en llamadas al objeto local, el cual podrá “hablar”, con cualquier entidad que “hable” XML a través de HTTP.

Para efectos del cliente de un servicio, Microsoft.NET proporciona clases ‘proxy’ que acceden a un servicio, por simple lectura de la descripción del servicio en cuestión (realizada la descripción en el estándar WSDL), de modo que cuando el cliente haga una petición, el ‘proxy’ genera una petición HTTP y la envía al servidor, que dará la respuesta obtenida.

WSDL es utilizado para describir el servicio *web*, especificar su localización y describir las operaciones y métodos que él expone, o sea que es una biblioteca utilizada para describir el servicio. Pero no se requiere conocer este protocolo para construir y utilizar los servicios *web*, porque .NET ofrece herramientas que generan automáticamente las descripciones, así como leerlas y mostrar la información más relevante. Y una vez que la dirección de la descripción WSDL de un servicio es localizada, su integración en una aplicación es inmediata.

UDDI es un sistema que permite publicar y descubrir información a cerca de los servicios *web* que están desarrollando en todo el mundo, con una infraestructura necesaria para poder registrar servicios *web* o buscar otros utilizando descripciones estándar basadas en WSDL. Es

merced a este sistema que se puede acceder a las descripciones WSDL que permiten utilizar un servicio *web*.

El concepto de Objetos Intermedios (*middle agents*), fue propuesto para cubrir la necesidad de un mecanismo que avise, encuentre, fusione, use, presente, gestione y actualice los servicios de información de los agentes. Así que los agentes intermedios son entidades a las que otros agentes comunican sus capacidades y que no proveen ni solicitan servicios, desde el punto de vista de la transacción en cuestión.

La ventaja de los llamados objetos intermedios es que permiten a los MAS (*MultiAgents Systems*), operar de modo seguro, aun contando con la aparición, movilidad y desaparición de los agentes.

Los agentes que cumplen con la definición de agentes intermedios, son diversos:

**Facilitadores:** A los que otros agentes subordinan su autonomía, a cambio de servicios. Los agentes Facilitadores, pueden coordinar actividades de los agentes y satisfacer peticiones en beneficio de sus agentes subordinados

**Mediadores:** Explotan el conocimiento codificado para crear servicios de más alto nivel a disposición de las aplicaciones.

**Brokers:** Reciben peticiones y realizan acciones, usando servicios de otros agentes en conjunción con sus propios recursos.

**Emparejadores (*Matchmakers*)** y páginas amarillas: Asisten a los que solicitan un servicio para buscar un proveedor de servicios, mediante las capacidades comunicadas anteriormente.

**Pizarras:** Repositorio de agentes que recibe y trata peticiones de proceso para otros agentes.

## 5. Los servicios de *web* semánticos

La más reciente y revolucionaria tecnología está representada por los servicio *web* autónomos. El servicio *web* transforma una colección estática de páginas de proveedores de servicios dinámicos que buscan información en la *web* automáticamente.

Los agentes que utilizan *web services*, se descubren y actúan recíprocamente entre sí, permitiendo una automaticidad entre ellos, que reduce la actividad humana y da potencia a esta tecnología. En consonancia con lo dicho, el *web service* ha de ser autónomo y escalable, necesitando una infraestructura multiagente

importante y potente, que ha de poseer la cualidad de permitir el análisis semántico del servicio, a fin de que pueda descubrir servicios, permitiendo varios tipos de interacción, incluidas las sincronías, asíncronas, *brokered*, (p2p) y *peer to peer*.

En realidad, los agentes *brokers* son más vulnerables a ciertos problemas, pero también pueden reaccionar más rápidamente, y cada *Middle Agent* tiene un tipo de forma de actuar, por su robustez y adaptabilidad, en relación con el ambiente externo y con los propios agentes.

El mecanismo *Agent-to-Agent Discovery* (A2A) permite a las aplicaciones de multiagente comunicarse, para actuar entre sí, recíprocamente, sobre conexiones típicas P2P de la red. Con el uso de *Agent-to-Agent* las aplicaciones pueden interactuar con otros agentes o clientes, en cualquier parte del mundo.

De este modo, los procesos automáticos, facilitan las interacciones del agente, descubriendo otros grupos de agentes dinámicamente, con intereses similares, que estén activos en la red, formando comunidades entrelazadas.

Con *Agent Local Area Discovery* los agentes y la infraestructura de servicios pueden encontrarse. *Discovery* activa los servicios entre agentes, services y servidores, para autoconfigurarse y facilitar así la interacción, lo que evita tener que realizar la configuración manual del sistema. Los servidores pueden localizarlos, informar y coordinar su funcionamiento. Los clientes pueden aprender de los nuevos servicios y automáticamente registrar la existencia de un nuevo servicio. Los agentes y los servicios de infraestructura pueden realizar acciones avanzadas, que respondan al entorno cambiante. Asimismo, los agentes pueden ser programados para sobrevivir en entornos incompletos.

## 6. Computación evolutiva

La computación evolutiva, tiene como objetivo fundamental, la búsqueda. En informática, los algoritmos de búsqueda, definen los problemas en términos de un espacio de búsqueda o un espacio solución, formado por todos los posibles elementos que cumplen con las características y/o objetivos propuestos [Kanal 1988]. El problema de optimizar ciertos valores de una función, se

convierte en buscar dichos valores dentro del espacio de búsqueda.

Los algoritmos evolutivos de búsqueda, se inspiran en y por la evolución de la naturaleza. Estos algoritmos, buscan una solución óptima, mediante la evolución de las soluciones anteriores, por lo tanto, en lugar de trabajar con una solución a la vez, estos algoritmos, consideran una gran población de soluciones al mismo tiempo, que debe evolucionar hacia la mejor solución.

### 7. Objetos de aprendizaje

Las tecnologías relacionadas con la enseñanza basadas en tecnología sustentan el aprendizaje en el empleo de piezas de grano fino de las que están formados los cursos o experiencias educativas: los llamados objetos didácticos o *learning objects*. Toda experiencia educativa, está formada por un conjunto de actividades de enseñanza cuyo soporte son recursos de aprendizaje electrónicos, entendiéndose por ello cualquier representación de una información que se utiliza para llevar a cabo dichas experiencias educativas. El proceso de creación y puesta a disposición de los usuarios de una experiencia educativa implica la creación, descubrimiento y agregación de —generalmente varios— objetos didácticos simples para conformar recursos educativos más complejos que tengan sentido. Es el uso de estas pequeñas piezas y la posibilidad de ensamblarlas a voluntad para construir con ellas modelos agregados de estructura superior al estilo de los bloques LEGO una de las características más atractivas de los entornos de enseñanza basados en estas tecnologías.

Ahora bien, si se ha de ser capaz de ensamblar los objetos didácticos para conformar una experiencia educativa, éstos deberían ser fácilmente accesibles e independientes del contexto en que se utilizan, y por tanto reutilizables. Lo ideal es que los recursos educativos se construyan como componentes reutilizables normalizados, lo que beneficia tanto a los desarrolladores de material educativo, como a los gestores de los mismos y a los propios alumnos. En los esfuerzos para desarrollar estándares sobre la creación y utilización de objetos didácticos, se han involucrado numerosas organizaciones e instituciones entre las que cabe destacar el Departamento de Defensa de los

Estados Unidos mediante la iniciativa ADL, el IEEE mediante el grupo de trabajo LTSC, el Consorcio para el aprendizaje IMS, el Aviation Industry Computer Based Training Committee (AICC) o ISO por citar sólo algunos de los más relevantes. Dichos esfuerzos de estandarización, han tenido como objetivo el dar forma al concepto de recurso educativo como objeto didáctico reutilizable verdadera piedra angular de la enseñanza basada en tecnología.

Se han alcanzado pues progresos significativos en el ámbito de la normalización, siendo algunos de los más importantes los referentes al intercambio de recursos educativos entre diferentes sistemas: el estándar de metadatos para objetos didácticos conocido como LOM, que especifica un esquema de datos conceptual para definir la estructura de una instancia de metadatos para un objeto didáctico. El propósito es facilitar la búsqueda, evaluación, adquisición y uso de los objetos didácticos por parte de los alumnos, profesores o sistemas automatizados, facilitando así el compartir e intercambiar recursos educativos, y permitiendo el desarrollo de catálogos e inventarios.

Existen otros trabajos en curso que pretenden cubrir otras áreas del proceso de *e-learning*, como el borrador para un estándar sobre los registros de información sobre los alumnos simplificando así el intercambio de datos entre sistemas cooperantes: PAPI Learner. Pero sin duda el esfuerzo más importante y que más interés suscita es SCORM, modelo común de objetos didácticos basado en componentes, cuyo principal objetivo es facilitar que los contenidos educativos estándar puedan compartirse entre diferentes sistemas. SCORM engloba varios estándares y especificaciones como el ya mencionado estándar de metadatos para objetos didácticos LOM, las especificaciones de IMS sobre secuenciación de contenidos y otras. Mediante la formalización y catalogación del contenido de un objeto didáctico mediante un registro de metadatos que acompaña al objeto didáctico, se permite que sea comprensible y asimilable por un programa, el cual pueda escoger en función de dichos metadatos el objeto didáctico que considere más adecuado, bien para integrarlo en una unidad de conocimiento superior o bien para mostrarlo y entregarlo al alumno final de cara al cumplimiento

de unos objetivos educativos previamente establecidos.

Los esfuerzos citados se dirigen a potenciar la reusabilidad del material educativo generado mediante la utilización de los estándares emergentes y con ello la eficiencia de los sistemas que los utilizan. Los beneficios derivados de la estandarización son evidentes: portabilidad entre diferentes plataformas, accesibilidad de los objetos didácticos, posibilidad de compartir y reutilizar los recursos educativos, durabilidad de los mismos con respecto a nuevas plataformas hardware y sistemas operativos futuros, etc.

En definitiva, el diseño de materiales educativos mediante el empleo de objetos didácticos debe considerarse como una opción muy seria ya que los materiales así diseñados tienen un potencial de reusabilidad muy importante, con los consiguientes beneficios que esto conlleva.

## 8. El paradigma del aprendizaje

Existen muchas aproximaciones al aprendizaje automático, entre ellas se pueden destacar:

**Aprendizaje supervisado:** es la forma más común de aprendizaje, a veces también se le llama programación mediante ejemplos. El individuo aprende mediante la visualización de ejemplos del problema, así como las soluciones deseadas. A partir de ese momento, el individuo hará predicciones basadas en las entradas, y si la salida difiere de la salida esperada, el agente se ajusta o adapta para producir la salida correcta. Este proceso se repite, hasta que el agente obtiene un grado suficiente de aciertos.

**Aprendizaje no supervisado:** este tipo de aprendizaje se usa para aquellos casos en que el individuo tiene que reconocer similitudes entre las entradas, o identificar ciertas características. Se le proporcionan los datos al individuo, y este se adapta, para lograr particionar los datos en grupos. Este proceso continúa hasta que los grupos se vuelven estables, y los mismos datos siempre son colocados en los mismos grupos.

**Aprendizaje por refuerzo:** Este tipo de aprendizaje, es un aprendizaje supervisado, cuando no están disponibles los datos para el entrenamiento. Es útil, cuando solo se dispone de los datos de entrada, y la solución solo puede ser conocida a posteriori. Este proceso de identificar la relación entre unos datos de entrada y su

posterior salida o resultado, se suele llamar asignación de crédito temporal. La desventaja de funcionar de esta manera, es que el proceso de aprendizaje, suele ser mucho más lento.

## 9. Arquitectura del conocimiento

Una organización es una red de conocimiento. Dicha red siempre ha existido. La cantidad y calidad de los enlaces va a estar relacionada con el aumento de los beneficios de la empresa. Por lo tanto es importante mejorar las relaciones, para que sean eficientes y productivas. La tecnología permite el envío, el personal permite el manejo de información, cuando se realiza una acción basada en esa información, esto permite el conocimiento. Basándose en este hecho se crea una arquitectura para que facilite el movimiento y transmisión del conocimiento.

El esquema utilizado de esta arquitectura es el que se describe a continuación. El primer nivel es el que corresponde al acceso del usuario a los distintos niveles. En este nivel se van a tener en cuenta los niveles de seguridad y de autenticación.

La interfaz que normalmente es un navegador, corresponde al segundo nivel. Este componente enmascara la complejidad de la red de la organización y los procesos que están situados por debajo utilizados para el envío de la información. La interfaz proporciona una vista universal de los documentos, correo electrónico, personal, etc.

El tercer nivel es el de la inteligencia – productos y programación que proporcionan un filtro de información, búsqueda a lo largo de distintos repositorios, personalización y agentes que conocen y afectan a las preferencias de usuario.

El cuarto nivel agrupa distintas aplicaciones que proporcionan el “valor añadido”. Estas aplicaciones y tecnologías proporcionan a los usuarios mejoras en la productividad y ayudan a indicar los caminos que facilitan la realización del trabajo. Este nivel incluye herramientas de publicación, análisis de sitios, gestión de documentos, bases de datos de discusión, bases de conocimiento de inteligencia competitiva, calendarios, balance de cuentas para ejecutivos,...

El nivel de transporte o de red se incluye para el desarrollo Web Intranet, correo electrónico, tecnologías de flujo y herramientas de colaboración.

El sexto y último nivel están agrupados todos los repositorios: los *Data Warehouse*, sistemas legales, repositorios de documentos, etc.

Una arquitectura de este tipo es adecuada también para la construcción de sistemas de tutorización en *e-learning*, puesto que permite reunir en una totalidad integrada cada una de las tecnologías necesarias que se han descrito.

### 10. Conclusiones

La conjunción de las tecnologías de la Ingeniería *Web* con las arquitecturas de sistemas de aprendizaje dentro de sistemas de agentes diseñados con el propósito de mejorar las capacidades pedagógicas de las aplicaciones de *e-learning* ha resultado ser muy fructífera tal como muestra el sistema descrito en este trabajo.

Este sistema está siendo evaluado en el ámbito del Grupo de Investigación de la Universidad Pontificia de Salamanca y en un corto periodo de tiempo estarán listos los primeros resultados.

### Referencias

- [1] Jennings, N.R. (1999) Agent – Based Computing: Promise and Perils. Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99).
- [2] Bentley P. (1999) Evolutionary Design by Computers. Morgan Kaufman.
- [3] Sanjuan y Sáez (2003) Genetic Algorithms for the Generation of Models with Micropopulations. Proceedings of evalife 2003. Springer
- [4] Sanjuán, O., Cejudo, V., Parra, J., Castillo, A., y Joyanes, L., (2003) Development of an application to support WEB navigation. Proceedings of ICWE 2003. Springer
- [5] Castillo, A., Sanjuán, O., Sáez, Y., y Joyanes, L., (2001) Approaching an Object-Oriented Methodology for Building Software Agents in Knowledge Management. Proceedings of SOCCO 2001.
- [6] Durfee (1988) Coordination of Distributed Problem Solvers. Boston, Kluwer Academic.
- [7] Durfee E. H., Lesser V. R. (1991) “Global Partial Planning: A Coordination Framework for Distributed Hypothesis”. IEEE Transactions on Systems, Man and Cybernetics. 21(6).
- [8] Ferber, J., (1999) Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence, Addison-Wesley.
- [9] Gasser L., (1992) “An Overview of DAI”, en Distributed Artificial Intelligence: Theory and Praxis, editado por N. M. Avouris y L. Gasser, Kluwer.
- [10] Gasser L., Hill R. W. (1990) “Coordinated Problem Solvers”. Annual Reviews of Computer Science. 4, pp 203-253.
- [11] Hewitt C. (1986) “Offices are Open Systems”. ACM Transactions of Office automation Systems. 4(3), pp. 271-287.
- [12] Hewitt C. (1977) “Viewing control Structures as Patterns of Messenger Passing”, Artificial Intelligence. Vol 8 nº 3, p 323-374.
- [13] Hoare C. A. R. (1985) Communicating sequential processes. Prentice Hall International.
- [14] Huhns M., Bridgelan D. (1991) “Multagent Truth Maintenance”. IEEE Transactions on Systems, Man and Cybernetics. 216 (6), pp. 1437-1445.
- [15] Kogan D. (1993) “Design and implementation of CB lite”. Proc. Conf. Organ. Comput.Syst, Milpitas, CA.
- [16] Kornfeld W. A., Hewit C.E. (1981), “The Scientific Community Metaphor”. IEEE Transactions on Systems, Man and Cybernetics. 11 (1) pp. 24-33.
- [17] Lesser V.R. (1991) “A Retrospective View of FA/C Distributed Problem Solving”. IEEE Transactions on Systems, Man and Cybernetics. 21 (6) 1347-1363.

- [18] Lesser V.R., Corkill D. D. (1981), "Functionally Accurate, Cooperative Distributed Problem Solving". IEEE Transactions on Systems, Man and Cybernetics. 11 (1) 81-96.
- [19] Liu J., Sycara K. (1997) "Coordination of Multiple Agents for Production management" Annals of Operations Research 75, pp. 235-289.
- [20] Loui R. (1987) "Defeat among Arguments: A system of Defeasible Inference" Computational Intelligence 3(1) pp. 100-106.
- [21] Malone R. W. (1988) "GAT is coordination theory?". National Science Foundation Coordination Theory Workshop. MIT.
- [22] Mason C., Johnson R. (1989) "DDDATMS: A Framework for Distributed Assumption-Based Reasoning" Distributed Artificial Intelligence vol II Editores M. Huhns y L. Gasser, pp. 293-318. San Francisco. Morgan Kaufmann.
- [23] Mataric M.J. (1992) "Designing Emergent Behaviors : from local Interaction to Colective Intelligence". From Animals to Animats 2, Proc. Of the Second International Conference Simulation of Adaptive Behaviour. Hawaii. Editores J-A Meyer, H.L. Roitblat y S.W.Wilson pp. 432-441, MIT Press.