



# Inteligencia Artificial

Revista Iberoamericana de Inteligencia Artificial

Iberamia (ed). ISSN 1137-3601 (print), 1988-3064 (on line). DOI: 10.4114/intartif

[Home](#)

[Current](#)

[Archives](#)

## Inteligencia Artificial. All issues from Vol.1, No.1 to Vol.14, No. 48

<a href="#">2010</a>	<a href="#">2009</a>	<a href="#">2008</a>	<a href="#">2007</a>	<a href="#">2006</a>
<a href="#">2005</a>	<a href="#">2004</a>	<a href="#">2003</a>	<a href="#">2002</a>	<a href="#">2001</a>
<a href="#">1999</a>	<a href="#">1998</a>	<a href="#">1997</a>		

### Vol 14, No 48 (2010)

#### Regular Papers

Guest Editorial: Fifth Latin American Workshop on NonMonotonic Reasoning 2009, (LANMR'09)

*Mauricio Osorio Galindo, Claudia Zepeda Cortés*

Modelling autonomic dataspace using answer sets [PDF](#)

*Gabriela MontielMoreno, Genoveva VargasSolar, José Luis ZechinelliMartini*

Efficient Computation of the Degree of Belief for a Subclass of Two Conjunctive Forms [PDF](#)

*Guillermo De Ita, Carlos Guillén*

Functional first order definability of LRTP [PDF](#)

*José Raymundo Marcial Romero, José Antonio Hernández*

A Lower Bound for Answer Set Solver Computation [PDF](#)

*Stefania Costantini, Alessandro Provetti*

### Vol 14, No 47 (2010)

#### Regular Papers

Multimedia application profiles based in standards: an specific one for UNED [PDF](#)

*José Luis Delgado Leal, Covadonga Rodrigo San Juan*

A robust algorithm for forming note complexes [PDF](#)

**Vol 2, No 6 (1998)****Special Issue: Monografía: Inteligencia Artificial Distribuida y Sistemas Multiagentes**

Inteligencia Artificial Distribuida y Sistemas Multiagentes [PS](#)

*A. GarciaSerrano, S. Ossowski*

Metodologías orientadas a agentes [PS](#)

*C.A. Iglesias, M. Garijo, J.C. González*

Inteligencia Artificial Distribuida en Entornos de Tiempo Real [PS](#)

*E. Vivancos, L. Hernandez, V. Botti*

Negociación mediante argumentación en sistemas multiagente [PS](#)

*C. Sierra, N. Jennings, P. Noriega, S. Parsons*

La coordinacion en Sociedades Artificiales de Agentes [PS](#)

*S. Ossowski, A. GarciaSerrano*

Interaccion entre agentes: Utilidad, Coaliciones y Negociacion [PS](#)

*M.V. Belmonte, J.L. Pérez de la Cruz, F. Triguero*

Subastas y Sistemas Multiagente [PS](#)

*P. Noriega, C. Sierra*

---

**Vol 2, No 5 (1998)****Special Issue: Monografía: Computacion Evolutiva**

Editorial: Computación Evolutiva. [ZIP](#)

*E. Alba, C. Cotta, F. Herrera*

Un estudio de la potencia computacional y robustez de los algoritmos evolutivos paralelos [ZIP](#)

*C. Cotta, E. Alba, J.M. Troya*

Algoritmos genéticos con reducción de varianza [ZIP](#)

*P. Cuesta, B. Galván, D. Greiner, G. Winter*

Computación evolutiva basada en un modelo de codificación implícita [ZIP](#)

*F. Veredas, F. Vico*

Algoritmos genéticos para el entrenamiento de redes de neuronas artificiales (RNA) recurrentes con activaciones temporales [ZIP](#)

*A. Pazos, J. Dorado, A. Santos, J.R. Rabuñal, N. Pedreira*

Reconocimiento de funciones de producción láctea mediante redes neuronales podadas y entrenadas con un algoritmo genético [ZIP](#)

*C. Hervás, S. Ventura, J. Martínez*

# Metodologías orientadas a agentes

Carlos A. Iglesias \*  
DTSCIT - ETSIT  
Univ. de Valladolid  
Real de Burgos s/n  
47011 Valladolid (España)  
cif@tel.uva.es

Mercedes Garijo  
DIT - ETSIT  
Univ. Pol. de Madrid  
Ciudad Universitaria s/n  
28040 Madrid (España)  
mga@gsi.dit.upm.es

José C. González  
DIT - ETSIT  
Univ. Pol. de Madrid  
Ciudad Universitaria s/n  
28040 Madrid (España)  
jcg@gsi.dit.upm.es

## Resumen

La tecnología de agentes está recibiendo una gran atención en los últimos años y, como consecuencia, la industria está comenzando a interesarse en adoptar esta tecnología para desarrollar sus propios productos. Sin embargo, a pesar del rápido desarrollo de teorías, arquitecturas y lenguajes de agentes, se ha realizado muy poco trabajo en la especificación (y aplicación) de técnicas para desarrollar aplicaciones empleando tecnología de agentes. La introducción de la tecnología de agentes en la industria requiere de metodologías que asistan en todas las fases del ciclo de vida del sistema de agentes. La necesidad de estas técnicas será especialmente necesaria a medida que el número de agentes de los sistemas aumente. En este artículo se presentan los principales trabajos que han propuesto el desarrollo de una metodología orientada a agentes, y la relación de las metodologías orientadas a agentes con las metodologías orientadas a objetos (OO) y las metodologías de ingeniería del conocimiento (ICO).

## 1 Introducción

La tecnología de agentes ha recibido gran atención en los últimos años y, como resultado, la industria está comenzando a interesarse en esta tecnología para desarrollar sus propios productos. Grandes empresas como IBM, Microsoft, Mitsubishi o BT cuentan ya con laboratorios de agentes inteligentes y han sacado los primeros productos al mercado, principalmente de agentes de usuario e internet. Sin embargo, para que el paradigma de la programación orientada a agentes se extienda en el desarrollo de aplicaciones de forma habitual, es necesario que se desarrollen técnicas de análisis y diseño con este paradigma.

La investigación en metodologías orientadas a

agentes es un campo incipiente. Debido a la relación del paradigma de la orientación a agentes con la orientación a objetos y con los sistemas basados en conocimiento, las metodologías orientadas a agentes no han surgido como metodologías totalmente nuevas, sino que se han planteado como extensiones tanto a metodologías orientadas a objetos como a metodologías de ingeniería del conocimiento.

En este artículo revisaremos por qué puede ser interesante tomar como punto de partida ambos enfoques (secciones 2 and 3), qué aspectos del paradigma de agentes no están incluidos en cada enfoque, y qué extensiones se han hecho en ambos campos. Finalmente, realizaremos una comparación entre las metodologías presentadas (sección 5).

---

\*Esta investigación ha sido realizada durante una visita del primer autor al Dep. de Ingeniería de Sist. Telemáticos de la Universidad Politécnica de Madrid.

## 2 Extensiones de metodologías OO

En esta sección revisaremos las siguientes metodologías orientadas a agentes: *Análisis y Diseño Orientado a Agentes* [?], *Técnica de modelado de agentes para sistemas de agentes BDI* [?] y *MASB*, (Método basado en escenarios multiagente), *Multi-Agent Scenario Based Method* [?, ?] y *Metodología Orientada a Agentes para Modelado de Empresas* [?].

### 2.1 Ventajas del enfoque

Podemos enunciar diversas razones que justifican la extensión de metodologías orientadas a objetos.

En primer lugar, hay notables *similitudes* entre el paradigma orientado a objetos y el paradigma de agentes [?, ?]. Desde los primeros tiempos de la IAD, fue establecida la estrecha relación entre la tecnología multiagente y la programación orientada a objetos concurrente [?, ?]. Tal y como enunció Shoham en su paradigma de programación orientada a agentes (AOP) [?], los agentes pueden considerarse como objetos activos, esto es, objetos con un estado mental. Ambos paradigmas comparten el paso de mensajes para comunicarse, y el empleo de herencia y agregación para definir su arquitectura. La principal diferencia [?] estriba en que los mensajes de los agentes tienen un tipo predeterminado (su acto comunicativo) y en la definición del estado mental del agente basado en sus creencias, intenciones, deseos, acuerdos, etc.

Otra posible ventaja proviene del *empleo habitual de los lenguajes orientados a objetos* para implementar sistemas basados en agentes porque se consideran un entorno natural de desarrollo [?, p. 34].

También se puede citar como ventaja la popularidad de las metodologías orientadas a objetos. Muchas metodologías orientadas a objetos han sido empleadas en la industria con éxito, como es el caso de OMT (*Object Modeling Technique*) [?], OOSE (*Object Oriented Software Engineering*) [?], OOD (*Object-Oriented Design*) [?], RDD (*Responsibility Driving Design*) [?] and UML (*Unified Modeling Language*) [?].

La experiencia del empleo de estas metodologías puede facilitar la adopción de metodologías de agentes que las extiendan porque por una parte, los ingenieros software pueden presentar una resistencia inicial a utilizar y aprender una metodología completamente nueva y, por otra parte, los gestores preferirán seguir metodologías que han sido previamente probadas y garantizan su aplicación. Además, la experiencia y difusión de las metodologías orientadas a objetos facilitarían el aprendizaje y comunicación de diagramas de las metodologías orientadas a agentes que las extiendan.

En cuanto a las tres vistas comúnmente empleadas para analizar un sistema en las metodologías orientadas a objetos también son interesantes para describir a los agentes: una *vista estática* para describir la estructura de los objetos/agentes y sus relaciones; una *vista dinámica* para describir las interacciones entre objetos/agentes; y una *vista funcional* para describir el flujo de datos entre los métodos/tareas de los objetos/agentes.

Por último, algunas de las técnicas para identificar objetos son propicias para ser aplicadas en identificación de los agentes. En concreto, tienen especial interés las técnicas de *casos de uso* [?] y *tarjetas CRC* (Clase-Responsabilidad-Colaboración) [?].

### 2.2 Aspectos no abordados

A pesar de las similitudes entre los paradigmas de la orientación a objetos y a agentes, obviamente, los agentes no son simplemente objetos. Por tanto, las metodologías orientadas a objetos no abordan [?, ?, ?] estos aspectos diferenciadores.

En primer lugar, los objetos tienen una estructura simple (atributos y métodos) mientras que los agentes tienen una estructura compleja (una arquitectura de agente). Una arquitectura de agente puede ser comparada a un módulo de las metodologías orientadas a objetos, y puede ser analizado e implementado siguiendo una metodología orientada a objetos.

Aunque tanto los objetos como los agentes emplean paso de mensajes para comunicarse, mientras que en los objetos el paso de mensaje se traduce en invocación de métodos, en los

agentes este paso de mensaje se suele modelar como el intercambio de un conjunto predeterminado de actos de habla, con protocolos asociados para negociar o responder a cada acto comunicativo. Además, los agentes realizan un procesamiento de los mensajes, y pueden decidir ejecutar o no la acción correspondiente al mensaje recibido.

Otro aspecto diferenciador consiste en que Los agentes pueden ser proactivos, reactivos y/o deliberativos, y este estado mental debe modelarse, caracterizando sus deseos, preferencias, objetivos, intenciones, . . . , frente a los objetos que simplemente son reactivos, limitándose a ejecutar un método cuando reciben un mensaje.

Uno de las principales diferencias radica en que los agentes se caracterizan por su estado mental, y las metodologías orientadas a objetos no proporcionan técnicas para modelar cómo los agentes llevan a cabo sus inferencias, su proceso de planificación, etc.

Otro aspecto diferenciador de los agentes frente a los objetos es su dimensión social. Es necesario definir procedimientos para modelar las relaciones “sociales” de los agentes.

## 2.3 Soluciones propuestas

**Análisis y diseño orientado a agentes de Burmeister** Este método [?] define tres modelos para analizar un sistema de agentes: el *modelo de agente*, que contiene los agentes y su estructura interna (creencias, planes, objetivos, . . . ); el *Modelo de organización*, que describe las relaciones entre los agentes (herencia y papeles en la organización); y el *modelo de cooperación*, describe las interacciones entre los agentes.

La metodología no fija un orden predeterminado para desarrollar los modelos. Los pasos para desarrollar cada modelo son:

**Modelo de Agente:** (1) identifica los agentes y su entorno empleando RDD. Burmeister propone una extensión de las tarjetas CRC para incluir creencias, motivaciones, planes y atributos de cooperación; (2) define motivaciones para cada agente (intereses, preferencias, responsabilidades, objetivos a largo plazo, etc.); (3) define la conducta de

cada agente empleando planes, con una notación gráfica similar a diagramas de transición de estados; (4) define el conocimiento y creencias necesarios para ejecutar cada plan.

**Modelo de Organización:** (1) identifica papeles y responsabilidades en el escenario; (2) construye una jerarquía de herencia, empleando notación OMT; (3) estructura los papeles en organizaciones y define las relaciones entre agentes (con notación OMT).

**Modelo de Cooperación:** (1) identifica las cooperaciones y los participantes en cada cooperación, e incluye la información de cada agente en una tarjeta CRC; (2) identifica los tipos de mensajes; (3) define los protocolos de cooperación para establecer los flujos posibles de mensajes entre los agentes cooperantes.

**Técnica de Modelado de Agentes para Sistemas de Agentes BDI** Este método [?] define dos niveles para modelar a los agentes:

- *La Vista Externa:* consiste en la descomposición del sistema en agentes y la definición de sus interacciones. Esto se lleva a cabo a través de dos modelos el *Modelo de Agente* y el *Modelo de Interacción*. El Modelo de Agente describe las relaciones jerárquicas entre las clases agentes (*Modelo de Clases de Agentes*) y las relaciones entre agentes concretos (*Modelo de Ejemplares de Agentes*). El Modelo de Interacción describe las responsabilidades, servicios e interacciones entre los agentes y los sistemas externos.
- *La Vista Interna:* modelado de cada clase de agente BDI a través de tres modelos: *el Modelo de Creencias*, que describe la información del entorno (conjunto de creencias) y el estado de creencias de cada agente (ejemplares concretos del conjunto de creencias); *el Modelo de Objetivos*, que describe los objetivos que un agente adopta o los eventos a los que responde; y *el Modelo de Planificación*, que describe los planes que un agente puede emplear para alcanzar sus objetivos.

El proceso de desarrollo de cada modelo de esta metodología consiste en los siguientes pasos:

**Vista Externa:** (1) Identifica los papeles (funcionales, organizativos, dependientes de la aplicación o dependientes de la implementación) del dominio de la aplicación para identificar los agentes y organizarlos en una jerarquía de clases de agentes (el Modelo de Clases de Agentes). Este modelo se representa con una notación tipo OMT; (2) Para cada papel del paso anterior, se deben identificar las responsabilidades asociadas y los servicios (actividades no descomponibles) proporcionados y empleados para cumplir con estas responsabilidades; (3) Para cada servicio, identifica las interacciones necesarias, los actos comunicativos de estas interacciones y el contenido de información. Identifica también la información requerida y los eventos; (4) Refina la jerarquía de agentes, teniendo en cuenta los detalles de implementación, y define clases de agentes concretas. Construye finalmente el Modelo de Ejemplares de Agentes.

**Vista Interna:** (1) Analiza las diferentes alternativas para alcanzar un objetivo. Descompón cada objetivo en subobjetivos y acciones, e itera sobre los subobjetivos. Los planes para responder a un evento o alcanzar un objetivo se describen empleando una notación gráfica similar a los diagramas de estados (statecharts) de Harel [?], extendiendo la notación para expresar fallos en un plan; (2) Construye las creencias de los sistemas. Asegúrate de que la información necesitada por los planes está incluida en los conjuntos de creencias. Los conjuntos de creencias no describen entidades reales, sino las creencias de un agente sobre estas entidades. Estos conjuntos de creencias se representan con notación OMT, y se deriva de ellos las funciones y predicados que se pueden aplicar a dichos objetos.

**Método basado en escenarios multiagente (MASB)** Este método [?, ?] propone una metodología para MAS que soporte trabajo cooperativo. La fase de análisis consta de las siguientes actividades:

- *Descripción de escenarios:* identificación con lenguaje natural de los principales papeles desempeñados por agentes software, agentes humanos y objetos del entorno. Descripción de los escenarios típicos.
- *Descripción funcional de los papeles:* descripción de los papeles de los agentes empleando diagramas de conducta, que describen el proceso desempeñado por cada papel de un agente, el acceso de los procesos a la información relevante (denominada acumulación), y las interacciones (denominadas flujos) con otros agentes. Estos diagramas pueden ser refinados en diagramas de transición, para detallar cada proceso y las relaciones entre los elementos.
- *Modelado conceptual de los datos:* modelado del conocimiento y los datos empleados por cada agente. Se proponen dos descripciones complementarias:
  - *Estructura conceptual de los datos:* atributos de las acumulaciones y relaciones entre ellas. La notación y la técnica empleada es seleccionada por el analista. Se proponen diagramas entidad-relación, diagramas de objetos, descripciones basadas en marcos, etc.
  - *Ciclo de vida de las entidades:* descripción de los estados permitidos para cada entidad.
- *Modelado del mundo:* descripción de los objetos del mundo con las mismas técnicas del paso previo.
- *Modelado de la interacción sistema-usuario:* simulación y definición de las interfaces de interacción hombre-máquina en cada escenario.

La fase de diseño consta de las siguientes actividades:

- *Descripción de los escenarios y de la arquitectura del MAS:* selección de los escenarios que van a ser implementados y de los papeles jugados por los agentes en estos escenarios.
- *Modelado de objetos:* refina el modelado del análisis, añadiendo jerarquías de herencia, atributos y métodos.

- *Modelado de agentes*: especificación de los elementos definidos en el paso de modelado conceptual de los datos en el análisis como estructuras de creencias. Se proponen dos descripciones complementarias:
  - *Estado de decisión de un agente*: descripción de cómo un agente alcanza sus objetivos. La notación gráfica consta principalmente de creencias, acciones, expectativas, planes, objetivos, mensajes e interacciones con otros agentes.
  - *Estado de acción*: descripción de los planes de un agente mediante una notación gráfica que expresa los estados y acciones del proceso de planificación.
- Modelado de las conversaciones.
- Diseño del sistema y validación global.

**Metodología orientada a agentes para modelado de empresas** Esta metodología [?] propone la combinación de metodologías orientadas a objetos (OOSE) y metodologías de modelado de empresas IDEF (*Integration DEfinition for Function modelling*) [?] y CIMOSA (*Computer Integrated Manufacturing Open System Architecture*) [?]. Los modelos propuestos son:

- *Modelo de funciones*: describe las funciones y sus entradas, salidas, mecanismos y control empleando diagramas  $IDEF_0$  que incluyen decisiones (selección de los métodos posibles dependiendo de la entrada y el control).
- *Modelo de casos de uso*: describe los actores involucrados en cada función, empleando notación de casos de uso OOSE. Los casos de uso se organizan en casos de uso abstractos para su reutilización.
- *Modelo dinámico*: este modelo se menciona para analizar las interacciones entre objetos. Los casos de uso se presentan en diagramas de trazas de eventos.
- *Sistema orientado a agentes*: se compone de las siguientes fases:

- *Identificación de agentes*: los actores de los casos de uso se identifican como agentes. Las funciones principales de un agente son sus objetivos y las posibilidades descritas en los diagramas  $IDEF_0$ .
- *Protocolos de coordinación*: se describen en diagramas de estados.
- *Invocación de planes*: se definen diagramas de secuencias que extienden los diagramas de trazas de eventos para incluir condiciones que indiquen cuándo un plan es invocado.
- *Creencias, sensores y actuadores*: las entradas de las funciones deberían ser modelados como creencias o obtenidos de los objetos mediante sensores, y los objetivos alcanzados deben ser modelados como cambios a las creencias o modificaciones a través de los actuadores.

## 3 Extensiones de metodologías ICO

### 3.1 Ventajas del enfoque

Las metodologías de ingeniería del conocimiento pueden proporcionar una buena base para modelar sistemas multiagente (MAS, *Multi-Agent System*) ya que las metodologías de ingeniería del conocimiento tratan del desarrollo de sistemas basados en conocimiento. Dado que los agentes tienen características cognitivas, estas metodologías pueden proporcionar las técnicas de modelado de la base de conocimiento del agente.

La definición del conocimiento de un agente puede considerarse un *proceso de adquisición de conocimiento*, y dicho proceso sólo es abordado por metodologías de ingeniería del conocimiento.

La extensión de metodologías de conocimiento puede aprovechar la experiencia adquirida en dichas metodologías. Además se pueden reutilizar las bibliotecas de métodos de resolución de problemas y ontologías así como las herramientas desarrolladas en estas metodologías.

Aunque estas metodologías han sido empleadas

en ámbitos más restringidos que las metodologías orientadas a objetos, también han sido aplicadas con éxito en la industria.

### 3.2 Aspectos no abordados

La mayor parte de los problemas planteados en las metodologías de ingeniería del conocimiento también se plantea, obviamente, en el diseño de sistemas multiagente: adquisición del conocimiento, modelado del conocimiento, representación y reutilización. Sin embargo, estas metodologías conciben un sistema basado en conocimiento como un sistema centralizado. Por tanto, no abordan los aspectos distribuidos o sociales de los agentes, ni en general su conducta proactiva, dirigida por objetivos.

### 3.3 Soluciones propuestas

Se han propuesto varias extensiones para desarrollar sistemas multiagente basándose en la metodología KADS o en su sucesora *CommonKADS* [?].

#### Extensión para sistemas cooperativos

Esta extensión [?] trata de ofrecer de la extensión de *CommonKADS* para sistemas basados en conocimiento cooperativos o sistemas multiagente. Las principales extensiones que se proponen son:

- *Modelo de organización*: proponen la inclusión de una carpeta más en los problemas de la organización, la carpeta de los problemas ya resueltos, para almacenar los problemas previamente resueltos que pueden ser utilizados por otros miembros. De esta forma, los problemas de la organización tendrían tres carpetas: problema actual, problemas con solución basada en conocimiento y problemas previamente resueltos.
- *Modelo de agente*: se extiende este modelo mediante la definición de un agente supervisor, que no realiza ninguna tarea del modelo de tareas, sino que se responsabiliza de la comunicación y cooperación entre el resto de los agentes: descomposición de tareas, asignación de tareas, control de

la comunicación, transformación de la representación del conocimiento y síntesis de conocimiento inconsistente.

- *Modelo de comunicación*: se propone el uso de diagramas de estado para modelar las comunicaciones.

#### El modelo de interacción de Kingston

Kingston [?] propone un modelo de interacción que es una variante del modelo de cooperación de KADS-I, cuyo sucesor fue el modelo de comunicación de CommonKADS (junto con las modificaciones al modelo de tarea para distribuir las tareas). En aplicaciones posteriores, este modelo de interacción ha sido abandonado y se ha empleado el modelo de comunicación.

Los pasos de desarrollo del modelo son:

- Dibujar una lista ordenada de las tareas que debe realizar el SBC.
- Identificar dependencias entre tareas (entradas/salidas) e interfaces con sistemas externos como bases de datos, usuarios o ficheros.
- Decidir qué tareas son realizadas por el SBC, cuáles por el usuario y cuáles de forma conjunta.

#### Adquisición de conocimiento basada en agentes para múltiples expertos

Dieng [?] propone un método de adquisición de conocimiento basándose en agentes para problemas en que existen múltiples expertos. Aunque no podemos considerar este método como una metodología de desarrollo de sistemas multiagente, ni fue este el propósito con que se desarrolló, la hemos incluido porque constituye un antecedente muy interesante de aplicación de KADS al modelado de agentes cognitivos.

El entorno de ese método son los problemas donde expertos en diferentes dominios cooperan para analizar un problema. En concreto, el caso de estudio es el diseño de un sistema experto para mejorar la seguridad de las carreteras, para lo cual se solicitó a dos psicólogos, tres ingenieros en infraestructuras de carreteras y dos ingenieros en vehículos que realizaran un análisis de

los accidentes de tráfico. Los psicólogos se centraron en el modelado cognitivo de los conductores y el diagnóstico de los errores de los mismos, mientras que los ingenieros en vehículos se centraron en los posibles fallos de los vehículos y las posibles soluciones para mejorar su diseño y construcción, y los ingenieros en infraestructuras de carreteras diagnosticaron los posibles peligros debido a la infraestructura de carreteras y los posibles modificaciones que deberían ser realizadas.

Analizando la conducta de los expertos, se comprobó que a veces un experto resuelve un problema sólo o solicita ayuda para resolver un problema concreto, o se ve obligado a resolverlo sólo porque sus compañeros están ausentes. Se pensó en desarrollar un sistema cooperativo que ayudara a los expertos y simulara el papel de sus compañeros si estos estaban ausentes.

La técnica de modelado con múltiples expertos se basa en la adquisición de conocimiento guiada por un modelo de agente cognitivo, y consiste en los siguientes pasos:

- Elicitación del conocimiento de los expertos. Las técnicas utilizadas son entrevistas individuales y colectivas que son transcritas para analizarlas posteriormente.
- Modelado de las múltiples experiencias. Se parte de los documentos obtenidos en las sesiones de elicitación y se guía el modelado con los modelos de KADS.

En concreto, partiendo de un análisis de las tareas que los expertos realizan, (por ejemplo análisis de los mapas, análisis de las fotos, que requiere analizar la posición de los vehículos, la carretera, analizar los choques, etc), se determina: qué tareas realizar cada experto por separado, qué tareas son realizadas por todos los expertos, qué tareas requieren la competencia de otro experto, cómo paliar los expertos la ausencia de otro experto, cómo cooperan los expertos, cómo evoluciona la cooperación entre los expertos y qué conflictos (p.ej. desacuerdos) pueden darse entre los expertos y cómo se resuelven.

El resultado de estas respuestas se plasma en cada modelo de agente, cuya plantilla se va rellenando. Este modelo consta de un modelo de la experiencia de KADS junto

con rasgos individuales (del agente independientes de la organización a que pertenece) y rasgos sociales (relativos a la organización a que pertenece y a las interacciones con otros agentes). Los rasgos individuales se clasifican en generales si son independientes del problema (dominio de competencia, objetivos de alto nivel, modelo de experiencia y recursos) y rasgos específicos del problema que dependen de la fase de la resolución del problema (estado, intenciones,...). Los rasgos sociales incluyen los modos de cooperación, los lenguajes de comunicación, los puntos de interacción y la estructura organizativa. Por ejemplo, en el problema de referencia, para el caso del agente psicólogo, se determinaría:

El modelo de la experiencia distingue entre nivel de tarea (análisis de la calidad de las entrevistas con los conductores y valoración de la fiabilidad de los conductores); nivel de inferencia (modelado y diagnóstico) y nivel de dominio (jerarquía de conceptos como conductor, vehículo, carretera, accidentes, etc.; jerarquía de relaciones, en concreto de influencia, p.ej. la edad repercute en la fiabilidad; y grafos de conocimiento sobre los tipos y conductas para tomar desvíos, estrategias de los conductores en los desvíos, evidencias que sugieren las hipótesis obtenidas de las entrevistas).

Los rasgos individuales adicionales son el dominio de competencia, aquí la psicología con especialización en desvíos en las carreteras).

Los rasgos sociales son los modos de cooperación (organización conjunta de la tarea colectiva y refuerzo de las hipótesis de otro agente); los puntos de interacción (consejos en las técnicas para entrevistar a los conductores y estrategias de los conductores para tomar un cruce); y el modelo de otros agentes (visión de la reconstitución cinemática efectuada por el ingeniero de vehículos y visión del conocimiento de los puntos que deben comprobarse en una infraestructura efectuada por el ingeniero en infraestructuras).

- Construcción del sistema basado en conocimiento cooperativo: tras analizar las especificaciones de los agentes, se debe determinar qué agentes se incluyen en el sistema cooperativo así como la descompo-

sición de la tarea global y la distribución de las subtareas entre el sistema basado en conocimiento y los usuarios. También debe especificarse si los agentes se pueden encuadrar en alguna estructura organizativa y cuáles son los puntos de interacción entre los agentes y las posibles respuestas. Se emplea además una técnica de comparación de grafos conceptuales para resolver conflictos potenciales entre los expertos, en todos los niveles del modelo de la experiencia: diferente terminología, diferente estructura de las tareas e inferencias, etc.

### **Método de diseño de sistemas multiagentes**

El método de diseño de sistemas multiagente [?] se propone facilitar la formalización del conocimiento y razonamiento de un sistema multiagente y consta de dos fases denominadas *Diseño Formal* y *Diseño Tecnológico* previas a la implementación y validación del sistema multiagente. El resultado de ambas fases es la definición de la arquitectura organizativa del sistema multiagente.

El Diseño Formal tiene como objetivo la estructuración del conocimiento de la aplicación y consta de tres fases:

- Identificación del conocimiento y razonamiento de la aplicación, empleando tipologías formales para estructurar el conocimiento y el razonamiento.
- Análisis del conocimiento y del razonamiento. Parte de la tipología de conocimiento y razonamiento obtenidas y desarrolla modelos conceptuales de conocimiento y razonamiento.
- Diseño de la arquitectura multiagente, definiendo las habilidades de cada agente según los modelos conceptuales.

La tipología para estructurar el conocimiento organiza el conocimiento en cuatro ejes:

- *Funcional*: distingue entre conocimiento figurativo (objetos), operativo (acciones), reflexivo (conocimiento sobre el conocimiento) y heurístico (conocimiento de cómo manipular el conocimiento)

- *Estructural*: asociaciones entre los elementos básicos de conocimiento. Pueden darse asociaciones intrínsecas (propiedades), contextuales (contexto de análisis del objeto o acción), composición (agrupamiento de elementos), taxonómico y causal.
- *Nivel de abstracción y específico del dominio*: clasificación de los elementos del nivel funcional según el nivel de abstracción y categorías dependientes del dominio.

La tipología para estructurar el razonamiento distingue entre razonamientos elementales que tratan sólo un tipo de conocimiento (p.ej. figurativo u operativo) y razonamientos híbridos, que combinan o integran varios razonamientos elementales.

El Diseño Tecnológico tiene como objetivo el análisis de la estructura del agente y de la sociedad de agentes y consta de tres fases:

- Identificación del razonamiento empleando la tipología tecnológica de los sistemas multiagente.
- Análisis del modelo de razonamiento partiendo de la tipología obtenida.
- Diseño de la arquitectura multiagente, definiendo las habilidades necesarias de cada agente.

La tipología tecnológica de los agentes distingue dos perspectivas: la perspectiva de agente y la perspectiva de grupo de agentes.

La perspectiva del agente analiza la estructura interna del agente y los tipos de razonamiento que emplea. La estructura del agente consta de cuatro partes: la comunicativa y reactiva, responsable de las comunicaciones con otros agentes y con el entorno; la racional, mecanismos de planificación y supervisión de tareas; la dependiente del dominio en que el agente es especialista; y la parte intencional que se describe el conocimiento sobre sí mismo y sobre el resto. Partiendo de esta estructura, se identifican los siguientes tipos de razonamiento:

- Razonamiento asociado a las comunicaciones: modos de intercambio de información, protocolos y contenido de los mensajes.

- Razonamiento asociado a la gestión de la base de conocimiento. Posibilidades de control del agente para acceder a su base de conocimiento, ejecutar tareas, solicitar información y delegar tareas.
- Razonamiento asociado a la gestión de recursos: habilidades del agente dependientes del dominio.
- Razonamiento intencional: análisis de sus habilidades para resolver problemas.

La perspectiva de grupo es analizada a través de tres parámetros: los modos de distribución de tareas y habilidades; los modos y protocolos de comunicación; y los modos de organización y comunicación.

**La metodología CoMoMAS** CoMoMAS (*Contribution to Knowledge Acquisition and Modelling in a Multi-Agent Framework*) [?] propone una extensión de *CommonKADS* [?] para modelar sistemas multiagente.

Propone los siguientes modelos:

**Modelo de agente** es el modelo central de la metodología. Podemos identificar dos elementos principales: la arquitectura interna del agente y el conocimiento que mantiene. Este conocimiento se clasifica en:

- *Conocimiento social*: conocimiento de la organización y de los papeles, creencias, intenciones y compromisos con otros agentes.
- *Conocimiento cooperativo*: describe los métodos posibles de cooperación y negociación con otros agentes, así como los protocolos y primitivas de comunicación. La descripción y uso de este conocimiento se detalla en el modelo de cooperación.
- *Conocimiento cognitivo*: engloba el conocimiento de resolución de los problemas junto con el conocimiento de las tareas, y se detalla en el modelo de la experiencia.
- *Conocimiento reactivo*: se compone de conductas primitivas y patrones de conducta, y se detalla en el modelo de la experiencia.

- *Conocimiento de control* para seleccionar los métodos de resolución de problemas, detallado en el modelo de la experiencia.

**Modelo de la experiencia** describe las competencias cognitivas y reactivas del agente. Se subdivide en:

- *Conocimiento de las tareas*: contiene información de la descomposición de las tareas, descrita en el modelo de tareas.
- *Conocimiento de resolución de problemas*: contiene información de los métodos de resolución de problemas (*Problem Solving Methods*, PSM) y de las estrategias, el conocimiento requerido para seleccionar un PSM.
- *Conocimiento reactivo*: contiene información de los procedimientos para responder a un estímulo.

**Modelo de tareas** las tareas se descomponen en subtareas. La única extensión es la distinción entre tareas resolubles por el usuario y tareas resolubles por un agente.

**Modelo de cooperación** el modelo de cooperación describe la cooperación entre varios agentes. Se descompone en métodos de resolución de conflictos (métodos de negociación y cooperación) y conocimiento de cooperación (primitivas, protocolos y terminología de interacción).

**Modelo del sistema** representa la organización de los agentes y describe la arquitectura del sistema multiagente y de sus agentes. Se descompone en aspectos arquitectónicos (heterogeneidad o no de la arquitectura del sistema y de la arquitectura de los agentes) y en aspectos organizativos (organización de los agentes; arquitecturas de los agentes; métodos de organización y conocimiento de la organización).

**Modelo de diseño** define cómo pasar de los modelos previos a código ejecutable. Recoge los requisitos funcionales y no funcionales de la aplicación, así como las guías de diseño (plataforma y lenguaje de implementación) y la historia del diseño.

El desarrollo de los modelos se realiza a través de un ciclo local de adquisición del conocimiento para desarrollar cada modelo y de un ciclo global de adquisición de conocimiento que describe los pasos para desarrollar el conjunto de modelos de la metodología.

Las actividades de modelado del ciclo global de adquisición de conocimiento son:

- *Análisis de requisitos*: su propósito es determinar los requisitos de diseño del sistema multiagente. Como resultado se obtiene una lista de requisitos indicando su importancia y sus interdependencias, que se almacenan en el modelo de diseño. Los pasos son:
  - Identificar los requisitos de diseño.
  - Identificar las interdependencias entre dichos requisitos.
  - Puntuar los requisitos según su importancia.
- *Análisis funcional*: su propósito es determinar las tareas que debe realizar el agente. El resultado debe ser una jerarquía de tareas resolubles por el usuario o por un agente, descritas en el modelo de tareas. Los pasos de este análisis son:
  - Analizar el dominio de aplicación y el espacio de estados del problema.
  - Determinar los objetivos, tareas y descomposiciones.
  - Determinar el flujo de datos y control.
  - Construir un modelo de tareas inicial y validarlo.
- *Análisis de competencia*: su propósito es identificar las competencias que el sistema debe tener para resolver los objetivos del modelo de tareas. Los resultados de este análisis se almacenan en el modelo de la experiencia. Los pasos son:
  - Determinar los métodos de resolución de problemas.
  - Determinar los recursos.
  - Determinar las estrategias.
  - Determinar las conductas.

- *Análisis social*: el propósito de este análisis es identificar las competencias sociales requeridas por los agentes. El conocimiento social facilita la cooperación e interacción de los agentes. El resultado de este análisis es la construcción del modelo del sistema. Los pasos son:

- Identificar las acciones y los objetivos conflictivos entre los agentes.
- Identificar las dependencias entre datos y objetivos.
- Determinar la organización de los agentes y los papeles asociados a cada agente.
- Determinar y asociar las intenciones con las acciones.
- Determinar y asociar los deseos con los objetivos.
- Determinar y asociar los compromisos con los objetivos.
- Determinar y asociar las creencias con los objetivos.

- *Análisis cooperativo*: el propósito del análisis cooperativo es identificar los protocolos de cooperación, métodos de cooperación y los métodos de resolución de conflictos entre los agentes. Los resultados se almacenan en el modelo de cooperación. Los pasos son:

- Identificar las necesidades y niveles de cooperación entre los agentes. Pueden ser dirigidos por los datos o por la ejecución.
- Identificar los conflictos en la interacción entre agentes, dirigidos por la ejecución.
- Determinar las estrategias de negociación y coordinación.

- *Construcción del agente*: el propósito de este paso es construir un conjunto inicial de modelos de agente. Esta construcción se basa en los requisitos de diseño y en las competencias requeridas. De hecho, supone identificar agentes partiendo de subárboles de la jerarquía de tareas y asignarles las competencias correspondientes. Los pasos son:

- Analizar el modelo de tareas para construir subárboles de subconjunto de tareas.
- Identificar los agentes basándose en categorías de tareas.
- Construir los agentes con conocimiento cognitivo y/o reactivo.
- Integrar el conocimiento de cooperación en los modelos de los agentes.
- Integrar el conocimiento social en los modelos de los agentes.

**MAS-CommonKADS** La metodología *MAS-CommonKADS* [?, ?, ?] extiende los modelos definidos en *CommonKADS*, añadiendo técnicas de las metodologías orientadas a objetos (OOSE, OMT) y de ingeniería de protocolos (SDL [?] y MSC96 [?]) para describir los protocolos entre agentes. La metodología comienza con una fase de conceptualización que emplea la técnica de casos de uso descritos con la notación de OOSE y formalizados con MSC (*Message Sequence Charts*) [?, ?]. Define los modelos descritos a continuación para el análisis y el diseño, que son desarrollados siguiendo un ciclo de vida en espiral.

**Modelo de Agente** describe las características de los agentes: su nombre, capacidades de razonamiento, sensores y actuadores, servicios, objetivos, etc. Este modelo incluye una plantilla textual para describir los agentes y varias técnicas para identificar los agentes como el análisis de los actores de la fase de conceptualización, análisis sintáctico del enunciado del problema, heurísticos para identificar agentes, reutilización de componentes (agentes) desarrollados previamente o empleo de tarjetas CRC extendidas para agentes.

**Modelo de Tareas** describe las tareas (objetivos) realizadas por los agentes y su descomposición, empleando plantillas textuales y diagramas para su desarrollo.

**Modelo de la Experiencia** describe el conocimiento que necesitan los agentes para poder conseguir sus objetivos. Se desarrollan varios ejemplares de este modelo para modelar inferencias sobre el dominio, razonamientos sobre el propio agente y sobre

otros agentes. Se distingue entre *métodos de resolución de problemas autónomos* que descomponen un objetivo en subobjetivos hasta llegar a objetivos directamente realizables y *métodos de resolución de problemas cooperativos* que descomponen un objetivo en objetivos que realiza el agente en cooperación con otros agentes.

**Modelo de Organización** describe tanto la organización en que va a ser introducido el sistema multiagente para estudiar la viabilidad del mismo como la organización de la sociedad multiagente. La descripción de la organización multiagente emplea una extensión de la notación gráfica del modelo de objetos de OMT, y describe la jerarquía de agentes, su relación con el entorno y su estructura.

**Modelo de Coordinación** describe las interacciones de los agentes, los protocolos empleados y las capacidades necesarias para realizar estas interacciones. El desarrollo de este modelo consta de dos hitos: la definición de los canales de comunicación entre los agentes, que permite el desarrollo de un primer prototipo, y el análisis de las interacciones para determinar qué interacciones son más complejas y requieren un protocolo de coordinación. Su desarrollo parte de la descripción de los escenarios prototípicos entre los agentes con notación MSC, en cuyos mensajes se especifica el acto de habla realizado y los datos intercambiados. El conjunto de interacciones permitidas se recoge en un diagrama de flujo de eventos. El modelado del procesado de los mensajes se realiza con diagramas de estados de SDL.

**Modelo de Comunicación** describe las interacciones hombre-máquina y los aspectos relevantes para desarrollar las interfaces de usuario.

**Modelo de Diseño** recoge los requisitos de los modelos previos y determina cómo se llevan a cabo. Se subdivide en tres submodelos:

**diseño de la red** que define los requisitos de la infraestructura de la red de los agentes [?] que son proporcionados por agentes de red. Se distinguen tres tipos de facilidades: *facilidades*

de red, tales como servicio de nombrado de agentes, páginas blancas y amarillas, protocolos de envío de mensajes disponibles, seguridad de red, contabilidad, etc.; *facilidades de conocimiento*, tales como servidores de ontologías y métodos de resolución de problemas, traductores entre lenguajes de representación del conocimiento, etc.; y *facilidades de coordinación*, que indican los protocolos primitivos de comunicación consensuados en la red, servidores de protocolos, facilidades de gestión de grupos, detección de malas conductas en el uso de recursos comunes, etc.

**diseño de los agentes** determinación de la arquitectura de agente más adecuada para cada agente, descomposición de cada agente en módulos recogiendo las funciones desempeñadas por cada agente de los modelos de análisis, y establecimiento de una correspondencia entre los módulos y la arquitectura seleccionada.

**diseño de la plataforma** selección y justificación del software y hardware empleado en el desarrollo del sistema multiagente

## 4 Técnicas de investigadores en MAS

La definición de metodologías para desarrollar MAS es relativamente reciente, tal y como se ha presentado. Sin embargo, los MAS han sido aplicados con éxito en diferentes campos empleando diferentes plataformas multiagente. Durante esta aplicación, algunos investigadores han tenido en cuenta una perspectiva de ingeniería software, aunque no pueda hablarse de una definición “formal” de una metodología. En este apartado resumiremos algunas de estas contribuciones.

**La experiencia de ARCHON** ARCHON [?, ?] es un entorno de desarrollo para MAS, que propone una metodología de desarrollo:

- *Análisis del problema multiagente*: compara dos perspectivas de desarrollo:
  - Descendente (*top-down*): el diseñador identifica (1) los objetivos del sistema; (2) las tareas necesarias para alcanzar dichos objetivos; (3) la descomposición de las tareas; y (4) flujo de datos entre tareas.
  - Ascendente (*bottom-up*): esta perspectiva permite el uso de sistemas preexistentes, restringiendo el análisis descendente.
- *Diseño de la comunidad de agentes*: consta de dos ciclos.
  - Diseño de la comunidad: (1) definir la granularidad de los agentes; (2) determinar el papel de cada agente y de la comunidad; (3) diseño de las interfaces de usuario; (4) asignación de habilidades a los agentes; y (5) enumeración de los mensajes intercambios.
  - Diseño del agente: codificación de las habilidades de cada agente (planes, reglas, etc.).

**La experiencia de MADE** MADE [?, ?] es un entorno de desarrollo para prototipado rápido de MAS. Propone una metodología de desarrollo para diseñar MAS que extiende los cinco estados para adquirir conocimiento enunciados por Buchanan [?]: Identificación, Conceptuación, Descomposición (añadido este paso para identificar los agentes), Formalización, Implementación y Prueba (aquí se añade la prueba de integración de los agentes).

También se establecen unos requisitos que una metodología orientada a agentes debería ofrecer: ayuda en la composición y descomposición de los agentes y bajo acoplamiento entre agentes y alta cohesión en cada agente.

**Otros trabajos** El tema del diseño de interfaces de usuario para IAD ha sido discutido en [?, ?]. Ninguna metodología aborda este tema de forma específica, si bien sí es propuesto en varias metodologías (Modelo de Interacción de Kinny, MASB y Modelo de Comunicación de MAS-CommonKADS).

El enfoque metodológico denominado *Cassiopeia* [?], aplicado al dominio de robots jugando al fútbol, distingue tres pasos para diseñar un sistema multiagente: (1) Listado de las conductas elementales de los agentes con técnicas funcionales u orientadas a objetos (p.ej. chutar, bloquear un oponente, etc.); (2) Analizar las conductas relacionales, es decir, estudiar las dependencias entre los agentes con un grafo de acoplamiento (p.ej. si chuto la pelota me ayudo a mí o a otro agente, si defendiendo puedo ayudar a coger la pelota del oponente, etc.); y (3) descripción de la dinámica de la organización, es decir, determinar quién puede comenzar una coalición analizando el grafo de acoplamiento (p.ej. en el fútbol la iniciara siempre el chutador) y cuándo puede decidirse terminarla.

Se han definido varios lenguajes de coordinación que pueden ser una alternativa para el modelado de las interacciones en las metodologías orientadas a agentes. Podemos destacar a *COOL* [?] y *AgentTalk* [?], que emplean máquinas de estado finito para representar las conversaciones y a *Yubarta* [?], que emplea la técnica de descripción formal *LOTOS* para modelar estas conversaciones.

Los lenguajes formales para definir agentes tales como *DESIRE* [?] podrían ser también empleados como lenguajes de diseño detallado. Algunas ventajas de su uso serían la validación y verificación de las propiedades dinámicas del sistema.

## 5 Conclusiones

Tras la revisión de las metodologías presentadas, podemos destacar que las mismas comparten la necesidad de modelar en el análisis y en el diseño algunos conceptos clave de los sistemas multiagente como son su estado mental, las tareas u objetivos que tiene realiza, las interacciones entre agentes y las relaciones de grupo entre agente. A continuación señalamos las posibles correspondencias y diferencias entre cada metodología revisada y la aquí propuesta.

En la propuesta de Kinny et al. la vista externa modela las relaciones entre agentes (nuestro modelo de organización) y las interacciones (nuestro modelo de coordinación). La vista en-

terna modela el estado mental de un agente BDI (nuestros modelos de agente y experiencia). El modelado de las interacciones sólo está enunciado y no se ofrece una notación especial para modelar los agentes ni sus relaciones (salvo las de herencia con notación OMT). El modelado del estado mental modela las creencias como objetos con notación OMT y propone un diagrama de planes con notación similar a los diagramas de estado de Harel, que podría verse como una alternativa al modelo de la experiencia, aunque éste ofrece un marco más elaborado.

Burmeister distingue entre un modelo de agente (nuestros modelos de agente y experiencia), un modelo de organización (nuestro modelo de organización) y un modelo de cooperación (nuestro modelo de coordinación). Los modelos sólo están enunciados con unas recomendaciones para su construcción, sin proponer ninguna notación. Propone una técnica de documentación de agentes basada en una extensión de las tarjetas con motivaciones, creencias y planes, similar a la propuesta en el modelo de agente.

La propuesta de Kendall combina técnicas orientadas a objetos con técnicas de modelado de empresas. Comparte con *MAS-CommonKADS* la identificación de agentes a partir de actores de casos de uso. Sin embargo, el desarrollo y el número de modelos es confuso en el artículo.

La propuesta de MASB cubre tanto el análisis como el diseño, ofreciendo notación gráfica para modelar el estado mental de los agentes, que hemos encontrado poco intuitiva, pero que puede ser una alternativa al modelo de la experiencia de *MAS-CommonKADS*. Los diagramas de conducta son similares a los de casos de uso, por lo que encontramos que serán más fáciles de aprender. Tampoco se cubre el modelado de las interacciones entre agentes.

CoMoMAS propone también la extensión de *CommonKADS* para sistemas multiagente, aunque con resultados muy diferentes. Su extensión se centra en el modelo de la experiencia, mientras que el resto de extensiones (organización, interacciones, identificación de agentes, etc.) apenas se desarrollan. En cuanto al modelo de la experiencia, en CoMoMAS se distingue explícitamente el conocimiento reactivo mientras que en *MAS-CommonKADS* se hace depender esta decisión de la arquitectura en

el diseño. La extensión de CML (*Conceptual Modelling Language*) propuesta en CoMoMAS podría ser adoptada en *MAS-CommonKADS* con pequeñas modificaciones.