

DEPARTAMENTO DE INFORMÁTICA Y AUTOMÁTICA  
FACULTAD DE CIENCIAS



**VNiVERSiDAD  
D SALAMANCA**

**TESIS DOCTORAL**

MODELO ADAPTATIVO PARA ORGANIZACIONES VIRTUALES DE  
AGENTES

**AUTORA**

D<sup>a</sup>. Sara Rodríguez González

**DIRECTORES**

Dr. D. Juan Manuel Corchado Rodríguez

Dr. D. Javier Bajo Pérez

Marzo de 2010



La memoria titulada “MODELO ADAPTATIVO PARA ORGANIZACIONES VIRTUALES DE AGENTES” que presenta D<sup>a</sup>. Sara Rodríguez González para optar al Grado de Doctor por la Universidad de Salamanca ha sido realizada bajo la dirección del profesor Dr. D. Juan Manuel Corchado Rodríguez, Profesor Titular del Departamento de Informática y Automática de la Universidad de Salamanca, y por el profesor Dr. D. Javier Bajo Pérez, profesor Encargado de Cátedra de la Escuela Universitaria de Informática de la Universidad Pontificia de Salamanca.

Salamanca, marzo de 2010

Los Directores

El Graduando

Fdo: Dr. D. Juan M. Corchado Rodríguez  
Profesor Titular de Universidad  
Informática y Automática  
Universidad de Salamanca

Fdo. D<sup>a</sup>. Sara Rodríguez González

Fdo: D. Javier Bajo Pérez  
Profesor Encargado de Cátedra  
Escuela Universitaria de Informática  
Universidad Pontificia de Salamanca



Sara Rodríguez González: *Modelo Adaptativo para Organizaciones Virtuales de Agentes*, PhD on Computers and Automation, © March 2010

Supervisors:

Dr. D. Juan Manuel Corchado Rodríguez  
Dr. D. Javier Bajo Pérez

Location:

Salamanca

*A mi abuela Nieves.... ¡Llegué!*



## RESUMEN

Cada vez gana más fuerza la idea de que modelar las interacciones de un sistema multi-agente no puede estar relacionado solamente con el propio agente y sus capacidades de comunicación, si no que es necesario utilizar los conceptos de una ingeniería organizacional. Es posible establecer distintos tipos de organizaciones de agentes dependiendo del tipo de comunicación, del modo de coordinación entre agentes, así como del tipo de agentes que conforman el grupo. Cada organización necesita de un soporte de coordinación que determine explícitamente cómo deben organizarse los agentes y llevar a cabo las acciones y tareas asignadas dentro de la misma. En este estudio se presenta un nuevo modelo de coordinación para una organización de agentes. La principal novedad del modelo consistirá en la capacidad de planificación dinámica y adaptativa para llevar a cabo una distribución óptima de las tareas de los agentes miembros de la organización. Se trata de un modelo único en su concepción, que permite dotar a una organización de capacidades auto-adaptativas en tiempo de ejecución para entornos altamente dinámicos. De esta forma, el comportamiento de un agente estará determinado por las metas que desee alcanzar, pero teniendo en cuenta el resto de metas de los agentes compañeros y los cambios en el entorno. Complementariamente el modelo es evaluado en un sistema multi-agente desarrollado dentro de la arquitectura orientada a organizaciones THOMAS y simulado en un entorno virtual.

## ABSTRACT

An idea that seems to be gaining considerable ground is that modeling the interactions of a multi-agent system cannot be related exclusively to the actual agent and its communication capabilities, but must involve the use of concepts found in organizational engineering as well. It is possible to establish different types of agent organizations according to the type of communication, the coordination among agents, and the type of agents that comprise the group. Each organization needs to be supported by a coordinated effort that explicitly determines how the agents should be organized and carry out the actions and tasks assigned to them. This thesis work presents a new global coordination model for an agent organization. The primary novelty of the model consists of the dynamic and adaptive planning capability to distribute tasks among the agent members of the organization as effectively as possible. This model is unique in its conception, allowing an organization in a highly dynamic environment to employ self-adaptive capabilities in execution time. This allows for the behavior of an agent to be determined by the goals it wishes to reach, while still giving consideration to the goals of other agents and any changes in the environment. The model is evaluated in a multi-agent system developed within an architecture oriented towards THOMAS organizations and simulated in a virtual environment.



## AGRADECIMIENTOS

*Durante la realización de esta tesis he recibido ayuda y apoyo de muchas personas a las que quisiera expresar mi más sincero agradecimiento.*

*En este trabajo se habla mucho de cooperación, de coordinación, de trabajo en grupo... Todos estos conceptos se ven reflejados también en el trabajo de fuera, en el que no está en los libros, ni en los artículos, ni en el ordenador, ni en las clases. Esta tesis no podría haberse llevado a cabo sin la ayuda de muchas personas.*

*En especial, dar las gracias a mi director, Juan M. Corchado, quien hace ya más de seis años me dió la oportunidad de entrar en este mundo y gracias al cual he vivido experiencias que nunca hubiera imaginado. Y a mi co-director, Javier Bajo, quien me ha guiado siempre con buenas ideas y con quien he compartido muy buenos momentos. Los dos me han enseñado a valorar el esfuerzo en el trabajo y me han ayudado y guiado siempre en el desarrollo de esta investigación.*

*Dar las gracias también al resto de compañeros del grupo BISITE, en especial a Fran y Dante, con los que he compartido muchos momentos desde el principio. A Carol, Bea, Oscar, Fer y Alberto sin los que los cafés de los descansos no serían posibles. Y a todos los demás, Deanna, Rosa, Yanira, Cristian, Emilio, Aitor, Chinchés, Manuel... A todos ellos tengo que agradecerles el llegar aquí.*

*También quiero dar las gracias a mis compañeros del departamento de Informática y Automática de la Universidad de Salamanca, algunos de los cuales me conocieron hace ya mucho tiempo empezando la carrera y con quien ahora comparto horas de trabajo.*

*Unas gracias mezcladas con un poco de perdón a mi familia y amigos. Sé que a veces he estado un poco insoportable. Os prometo volver a La Fuente e ir a veros a Madrid.*

*Quisiera hacer una mención muy especial a mis padres Merce y Heraclio y a mi hermano, Quique. Gracias por apoyarme y sobre todo por ayudarme a distinguir la cosas importantes de las que a lo mejor, no lo son tanto.*

*Por último, gracias a Antonio, por su apoyo incondicional en los buenos y malos momentos y por perdonarme cuando no tenía tiempo para nada. Sin tí no lo hubiera conseguido.*



---

---

# ÍNDICE DE CONTENIDO

Resumen.....	vii
Abstract.....	viii
Agradecimientos.....	ix
Índice de Contenido.....	1
Índice de Figuras.....	5
Índice de Tablas.....	8
Parte I. Introducción.....	9
1    Introducción.....	11
1.1    Descripción del problema, motivación e hipótesis.....	11
1.2    Objetivos.....	13
1.3    Metodología.....	14
1.4    Estructura de la memoria.....	15
Parte II. Estado del arte y enfoques.....	19
2    Agentes y Sistemas Multi-agente.....	21
2.1    El Paradigma de Agentes.....	21
2.1.1    ¿Qué es un agente?.....	21
2.1.2    Clasificación de los agentes.....	24
2.1.3    Arquitecturas.....	26
2.1.4    Agentes BDI.....	28
2.1.5    Planificación y agentes.....	32
2.2    Sistemas Multi-agente.....	39
2.2.1    Interacción entre agentes.....	41
2.2.2    Adaptación en SMA.....	43

## Índice

---

2.2.3	Sociedades de agentes .....	45
2.2.4	Tendencias .....	49
3	Modelos sociales para sistemas adaptativos y sus enfoques.....	52
3.1	Organizaciones de agentes.....	52
3.1.1	Conceptos organizativos.....	55
3.1.2	Arquitecturas.....	61
3.2	Coordinación .....	69
3.2.1	Modelos de coordinación.....	71
3.3	Adaptación.....	74
4	Tecnologías complementarias.....	79
4.1	Sistemas CBR y CBP.....	79
4.1.1	Planificación basada en casos.....	84
4.2	Agente CBR- BDI .....	87
4.3	Agentes virtuales inteligentes .....	91
	Parte III. Propuesta .....	95
5	Modelo de Organización.....	97
5.1	Motivación.....	98
5.2	Trabajos relacionados .....	99
5.3	Modelo de planificación adaptativa .....	100
5.3.1	Agentes CBP- BDI .....	101
5.3.2	Modelo de planificación para un agente CBP-BDI en una OV.....	105
5.4	Conclusiones Preliminares.....	119
	Parte IV. Evaluación y conclusiones .....	121
6	Caso de estudio.....	123
6.1	Introducción .....	124

---

---

6.1.1	Algunos trabajos relacionados.....	128
6.2	Contexto.....	130
6.3	TouristOrg.....	132
6.4	Ejemplo de planificación.....	141
7	Resultados experimentales, conclusiones y trabajo futuro.....	152
7.1	Resultados.....	152
7.2	Conclusiones.....	163
7.3	Contribuciones a la investigación.....	165
7.4	Trabajo Futuro.....	167
Parte V.	Model Overview.....	171
8	Model Overview.....	173
8.1	Introduction.....	173
8.2	State of the art.....	176
8.3	Proposed Model.....	179
8.3.1	CBP- BDI Agents.....	179
8.3.2	Planning model for a CBP-BDI agent.....	181
8.4	Case Study.....	188
8.4.1	Context.....	188
8.4.2	Intelligent virtual agents.....	191
8.4.3	Multi-agent environment for our planning model.....	192
8.5	Examples of planning.....	195
8.6	Results and conclusions.....	203
	Bibliografía.....	207
	Referencias bibliográficas.....	209
Anexo A	Análisis y Diseño del Caso de Estudio.....	229

## Índice

---

A.1	Gormas .....	234
A.2	Análisis del sistema .....	237
A.2.1	Fase A. Misión .....	237
A.2.2	Fase B. Tareas y procesos.....	243
A.3	Diseño de la estructura de la organización.....	250
A.3.1	Fase C. Dimensiones Organizativas.....	250
A.3.2	Fase D. Estructura Organizativa .....	252
A.4	Diseño de la dinámica de la organización.....	255
A.4.1	Fase E. Procesos de información-decisión.....	255
A.4.2	Fase F. Dinamicidad del sistema abierto .....	257
A.4.3	Fase G. Sistema de medición, evaluación y control.....	261
A.4.4	Fase H. Sistemas de recompensas .....	261
Anexo B	Herramientas Utilizadas .....	263
B.1	EMFGormas.....	263
B.2	THOMAS .....	264
B.3	REPAST .....	268
B.4	JADE.....	273
B.5	Herramientas de modelado .....	275
Anexo C.	Proyectos, Publicaciones y Trabajos Relacionados .....	276

# ÍNDICE DE FIGURAS

Figura 1. Clasificación de los agentes según su característica primaria [Nwana, 1995].....	26
Figura 2. Arquitectura básica de un agente BDI .....	30
Figura 3. Representación de plan como secuencia de acciones en planificación clásica.....	34
Figura 4. Topología de organización. De izquierda a derecha: jerarquía, holarquía, coaliciones.....	57
Figura 5. Topología de organización. De izquierda a derecha: grupos, congregaciones.....	57
Figura 6. Topología de organización. De izquierda a derecha: federaciones, organizaciones matriciales.....	57
Figura 7. Aprendizaje por facilitación social. Aprendizaje por imitación .....	60
Figura 8. Arquitectura THOMAS.....	63
Figura 9. Agente Virtual Inteligente.....	92
Figura 10. Modelo de planificación .....	106
Figura 11. Representación de la actuación del agente CBP-BDI ante una situación de interrupción.....	116
Figura 12. Esquema de funcionamiento del sistema.....	118
Figura 13. Nuevas tecnologías en el sector turístico.....	124
Figura 14. Diagrama del modelo de organización (vista funcional). Misión del sistema. 134	
Figura 15. Diagrama del modelo de organización, vista funcional, (funcionalidad externa).....	135
Figura 16. Diagrama del modelo de actividad del servicio Asignar Guía. a) Perfil del servicio b) Relaciones entre las A-Tareas que lo componen .....	136
Figura 17. Diagrama actualizado del modelo de organización, en el que se muestran los agentes internos, externos y los roles que juegan.....	137

## Índice

---

Figura 18. Modelos tridimensionales utilizados para representar físicamente en el entorno agentes del tipo monumento (Catedral de Salamanca, Iglesia de San Esteban, Casa Las Conchas).....	139
Figura 19. Módulos THOMAS.....	141
Figura 20. a) Representation of a space $\mathcal{R}^3$ for tasks . b) Representation of a space $\mathcal{R}^3$ for replanned tasks.....	148
Figura 21. Captura de pantalla del módulo de simulación .....	153
Figura 22. a) Número de agentes trabajando simultáneamente b) Eficiencia en la organización de agentes con y sin planificación .....	155
Figura 23. Tiempo (Ticks) vs. Número de agentes Guía y Turistas .....	156
Figura 24. Evolución del planificador .....	158
Figura 25. Diferentes enfoques para distribución de tareas.....	159
Figura 26. Planificación distribuida vs. Planificación central respecto al tiempo. 159	
Figura 27. Meta-modelos en MDD .....	233
Figura 28. Procesos entre metamodelos .....	234
Figura 29. Diagrama del modelo de organización (vista funcional). Misión del sistema. 240	
Figura 30. Diagrama del modelo de organización del sistema. a) Vista estructural. b) vista funcional, funcionalidad interna .....	245
Figura 31. Diagrama del modelo de organización, vista funcional, (funcionalidad externa) .....	246
Figura 32. Diagrama del modelo de actividad del servicio Asignar Guía. a) Perfil del servicio b) Relaciones entre las A-Tareas que lo componen .....	248
Figura 33. Diagrama del modelo de entorno del sistema.....	249
Figura 34. Diagrama del modelo de actividad. Relación entre servicios y objetivos funcionales.....	249
Figura 35. Árbol de decisión de la estructura organizativa.....	253



---

---

Figura 36. Diagrama actualizado del modelo de organización al adoptar una estructura de tipo congregación. a) Vista estructural, b) Vista funcional (funcionalidad externa). .....	254
Figura 37. Diagrama actualizado del modelo de entorno de la organización TouristOrg. 256	
Figura 38. Diagrama del modelo de interacción para el servicio "Consultar Guía" 257	
Figura 39. Diagrama de colaboración de la interacción Consultar Guía.....	257
Figura 40. Ejemplo de adquisición de rol por parte de agente externo en THOMAS 259	
Figura 41. Diagrama actualizado del modelo de organización, en el que se muestran los agentes internos, externos y los roles que juegan. ....	260
Figura 42. Interfaz EMFGormas e Eclipse .....	263
Figura 43. Estructura de plugins de Eclipse.....	264
Figura 44. Aplicación THOMAS .....	265
Figura 45. Interfaz THOMAS .....	266
Figura 46. Interfaz manejo de SF en THOMAS .....	267
Figura 47. Interfaz manejo de OMS en THOMAS.....	267
Figura 48. Interfaz manejo de agentes e interacciones en THOMAS.....	267
Figura 49. Interfaz manejo de servicios en THOMAS.....	268
Figura 50. Desarrollo de modelos en REPAST S.....	271
Figura 51. Interfaz de usuario en REPAST S .....	271
Figura 52. Simulación.....	272
Figura 53. Combinación JADE-REPAST.....	272
Figura 54. Diagrama de las principales clases del modelo de simulación implementado en Repast.....	273
Figura 55. Arquitectura JADE .....	274
Figura 56. Google SketchUp.....	275

## ÍNDICE DE TABLAS

Tabla 1.	Resumen de los servicios ofrecidos en THOMAS.....	67
Tabla 2.	Ciclo de un sistema CBR.....	90
Tabla 3.	Ejemplo de tarea.....	142
Tabla 4.	Atributos de los casos .....	157
Tabla 5.	Architectures comparison.....	162
Tabla 6.	Documento A.1 - Misión Organizativa del sistema.....	238
Tabla 7.	Documento A.2 Grupos de Interés del sistema.....	241
Tabla 8.	Documento A.3- Condiciones de Entorno del sistema.....	242
Tabla 9.	B.1. Tecnología esencial del sistema.....	243
Tabla 10.	Documento B.2 Tecnología de Unidad de Trabajo. Descripción de las características del servicio.....	246
Tabla 11.	Documento C. Dimensiones Organizativas del sistema .....	252
Tabla 12.	Comparación herramientas de simulación.....	268

# PARTE I. INTRODUCCIÓN



# 1 INTRODUCCIÓN

---

Frecuentemente los sistemas multi-agente (SMA) se conciben como sistemas software compuestos de múltiples entidades computacionales encapsuladas, embebidos en un entorno, e interactuando de forma autónoma e inteligente. En la comunidad científica todavía existen opiniones diferentes respecto a qué considerar comportamiento inteligente, autónomo o incluso a cuándo los módulos o componentes deben llamarse agentes. Sin embargo, está ampliamente aceptada la idea de que la capacidad de coordinación y planificación es un aspecto clave para los SMA. Aún así, este consentimiento suele desaparecer cuando se intenta dar una definición concreta de conceptos como coordinación, planificación o negociación. Este hecho no es sorprendente ya que estos términos son utilizados en una gran variedad de disciplinas. Los sociólogos observan el comportamiento de grupos de personas, identificando mecanismos de coordinación; y los economistas se interesan por la estructura y la dinámica de los mercados. El interés de la informática está determinado principalmente por la necesidad de diseñar unos mecanismos de coordinación eficientes entre entidades computacionales.

En esta tesis se presenta un nuevo modelo de coordinación para organizaciones de agentes. La principal novedad del modelo consistirá en la capacidad de planificación dinámica y adaptativa para llevar a cabo una distribución óptima de las tareas de los agentes miembro de una organización. Se trata de un modelo único en su concepción, que permite dotar a una organización de capacidades auto-adaptativas en tiempo de ejecución para entornos altamente dinámicos. De esta forma, el comportamiento de un agente estará determinado por las metas que desee alcanzar, pero teniendo en cuenta el resto de metas de los agentes compañeros y los cambios en el entorno. En resumen, el modelo presentado permite definir las acciones que una organización de agentes debe llevar a cabo, considerando los cambios que pueden producirse durante la ejecución de un determinado plan. Es una planificación adaptativa dentro de una organización de agentes.

---

## *1.1 DESCRIPCIÓN DEL PROBLEMA, MOTIVACIÓN E HIPÓTESIS*

---

No existen los estudios aislados. La investigación es en gran medida el reconocimiento de la validez de ideas y trabajos previos y su aplicación, uso y/o mejora en diferentes contextos. La importancia de lo ya hecho, de lo ya investigado queda patente en cada nuevo método, cada nuevo modelo o cada nueva teoría concebida. Esto es cierto para este trabajo. En él se ofrece una nueva idea junto con el contexto, motivación e hipótesis que la hicieron surgir. Se presenta una visión general de las teorías básicas y las áreas de investigación relacionadas con la investigación para ayudar a obtener una mejor

comprensión de los conceptos y las ideas propuestas posteriormente. El fin no es dar una visión completa de estos temas, sino desplegar los fundamentos sobre los que se basa esta investigación. En concreto, se explicará el paradigma que envuelve a los agentes y sistemas multi-agente así como el camino recorrido desde su origen y las tendencias que más auge están teniendo actualmente. Se describirán las ventajas y desventajas que proporcionan los agentes en problemas de muy distinta índole [Wooldridge, 2002]. Y dado que la teoría de agentes ha evolucionado siguiendo distintas tendencias, nos centraremos en los agentes que consideramos más adecuados para la resolución de problemas en entornos dinámicos, ya que son los que centran el foco de este estudio. En nuestro caso, trabajaremos con agentes que utilizan el modelo deliberativo BDI (basado en los conceptos de creencias, deseos e intenciones) [Bratman et al., 1998] y que son capaces de cooperar con otros agentes dentro de una "sociedad".

Debido a los avances tecnológicos de los últimos años, la "sociedad" en la que participa un sistema multi-agente necesita cada vez más satisfacer requisitos tales como distribución, evolución constante y flexibilidad para permitir la entrada y salida de miembros de la misma, la gestión adecuada de la estructura organizativa que la define, la ejecución de los agentes en multi-dispositivos y en dispositivos con recursos limitados, etc. Todos estos requisitos definen un conjunto de características que se pueden abordar mediante el paradigma de los sistemas abiertos y de las organizaciones virtuales. Paradigmas que se tratarán en este trabajo y que han sido concebidos como una solución para la gestión, la coordinación y el control del comportamiento de los agentes [Dignum 2006].

Las organizaciones virtuales [Ferber et al., 2004] son un medio de entender el modelado de sistemas desde un punto de vista sociológico. El modelo de las organizaciones virtuales desde un punto de vista empresarial se apoya en los principios de la cooperación entre empresas dentro de una red, y explota aquellos elementos diferenciadores que aportan flexibilidad, capacidad y rapidez de respuesta, dentro de una estrategia dirigida a satisfacer al consumidor [Schertler, 1998]. Resulta pues adecuado que los sistemas software vayan tomando en consideración esta forma de organización que explota de manera eficiente las nuevas tecnologías de la información y la comunicación, para mejorar de este modo su competitividad.

Aún así, dentro del desarrollo de organizaciones a nivel de agentes, encontramos un conjunto de requerimientos [Rodríguez et al., 2009] que hacen necesarios nuevos modelos sociales en los que la utilización de sistemas abiertos [Zambonelli et al., 2003] y adaptativos [Di Marzo et al., 2004] sea posible. Los sistemas abiertos se caracterizan por la heterogeneidad de sus participantes, la confianza limitada, objetivos individuales en conflicto y una gran probabilidad de disconformidad con las especificaciones [Giret et al., 2005]. La adaptación al entorno es un reto conocido en todas las

investigaciones más recientes [Zambonelli et al., 2004][Reitbauer et al., 2004][Weyns et al., 2004][Capera et al., 2003][Razavi et al., 2005]. Se pretenden sistemas software capaces de responder y actuar por sí mismos ante cambios en su entorno. Para ello se hace necesario disponer de teorías, modelos, mecanismos, métodos y herramientas que permitan desarrollar sistemas con capacidad de reorganización y que puedan adaptarse de esta forma a futuros cambios en su entorno.

*Por tanto, la hipótesis de partida es que es posible modelar un sistema eficiente para la gestión de sistemas abiertos a partir de un modelo de organizaciones adaptativo que facilite una planificación distribuida y dinámica.*

Aprovechando las ventajas que nos proporcionan las particularidades del desarrollo de SMA desde el punto de vista organizacional, y considerando las carencias existentes actualmente respecto a una planificación adaptativa para un modelo social cualquiera, se formalizará un modelo para la planificación dinámica y adaptativa en una organización de agentes. Mediante la combinación de tecnología multi-agente y de técnicas de planificación basadas en investigación operativa, se llevará a cabo una distribución óptima de las tareas de los agentes miembros de la organización.

---

## 1.2 OBJETIVOS

---

Cada vez gana más fuerza la idea de que modelar las interacciones de un SMA no puede estar relacionado solamente con el propio agente y sus capacidades de comunicación, si no que es necesario utilizar los conceptos de una ingeniería organizacional. Es posible establecer distintos tipos de organizaciones de agentes dependiendo del tipo de de coordinación, del modo de comunicación, así como del tipo de agentes que conforman el grupo. Cada organización necesita de un soporte de coordinación que determine explícitamente cómo deben organizarse los agentes y llevar a cabo las acciones y tareas asignadas dentro de la misma.

*El objetivo principal que se pretende conseguir es modelar un sistema de planificación capaz de adaptarse a los objetivos comunes de los agentes deliberativos que participen en una organización. Para alcanzar dicho objetivo, se hace necesario estudiar de forma detallada los conceptos y formalismos utilizados tanto por agentes deliberativos BDI como en las estructuras organizacionales de las sociedades de agentes y en la teoría de la planificación; además de diseñar un formalismo que permita la integración.*

Para la realización de este trabajo es necesario llevar a cabo otros objetivos más específicos que lleven a la consecución del objetivo final. Entre ellos:

- ❖ Realizar un estudio de las tendencias en el desarrollo de sistemas multi-agentes, metodologías de análisis y diseño, aplicaciones y evolución.
- ❖ Realizar un estudio sobre los SMA basados en organizaciones, modelos sociales, así como de los conceptos organizativos y de adaptación y coordinación.
- ❖ Realizar un estudio de las tecnologías complementarias, incluyendo agentes virtuales inteligentes, sistemas de razonamiento basados en casos y aplicación de estas tecnologías en los SMA .
- ❖ Proponer un modelo de planificación aplicable a la construcción de organizaciones de agentes adaptativas.
- ❖ Buscar mecanismos formales útiles en la resolución de problemas relativos tanto a la adopción de objetivos como a la selección de planes dirigidos a la consecución de esos objetivos por parte de una organización. Una organización de agentes debe tener la capacidad de generar planes para sus agentes miembro y debe perseguir objetivos en relación a sus propios intereses.
- ❖ Diseñar un modelo integrador en el que una arquitectura de agentes pueda ejecutar acciones basándose en un modelo de planificación social en una organización de agentes. El pilar del modelo será el razonamiento basado en casos, es decir, apoyándose en experiencias similares ocurridas en el pasado, en contextos similares, la organización podrá actuar.
- ❖ Realizar el estudio de problemas concretos de aplicación del modelo en organizaciones virtuales, que se desarrollen en entornos dinámicos y distribuidos, con constantes cambios.
- ❖ Evaluar empíricamente los resultados obtenidos en entornos reales de aplicación.

---

### 1.3 METODOLOGÍA

---

La metodología de investigación seguida en el desarrollo de este trabajo se estructura en seis actividades. La primera actividad ha consistido en un estudio del estado del arte. En esta etapa se ha realizado una revisión y descripción del estado actual de todos los conceptos relacionados con la teoría organizativa de agentes, las metodologías de análisis y diseño de organizaciones, las plataformas y entornos de desarrollo y simulación de SMA, sistemas de coordinación y adaptación en organizaciones y sistemas de razonamiento basados en casos.

La segunda actividad ha consistido en la identificación de las ventajas y desventajas de todas las posibilidades revisadas en la primera actividad, haciendo un estudio exhaustivo de las tecnologías elegidas, entre ella, la plataforma para el desarrollo de sistemas multi-agente abiertos basados en



organizaciones THOMAS [Carrascosa et al., 2009][Giret et al., 2009] y la metodología de análisis y diseño GORMAS [Argente, 2008].

La tercera actividad ha consistido en la formalización de un modelo de planificación capaz de integrarse en un agente deliberativo, de tal forma que dote a los agentes del modelo social propuesto de un mayor grado de autonomía, adaptación al entorno y de capacidad de reorganización.

En la cuarta actividad se ha propuesto la evolución de una arquitectura multi-agente que describe los módulos necesarios para construir organizaciones de agentes con capacidad de adaptación y coordinación y que garantiza la compatibilidad con los estándares existentes para sistemas multi-agente y con arquitecturas orientadas a organizaciones.

La quinta actividad ha consistido en la aplicación de los resultados obtenidos en las actividades dos, tres y cuatro a casos de estudio reales en entornos dinámicos. Se ha obtenido un caso de estudio dentro del sector turístico, en el que la generación de planes y la distribución de recursos de forma óptima siempre ha sido un reto conocido. En la aportación de este trabajo tiene un peso importante el caso de estudio diseñado. Sin embargo, el objetivo del mismo no es una disertación del comportamiento de un entorno turístico, sino la validación del modelo de planificación propuesto mediante la simulación de una organización dentro de este sector.

Finalmente, la sexta actividad realizada ha consistido en la evaluación de resultados y obtención de conclusiones. Las actividades cinco y seis pueden definirse en tres objetivos o metas a alcanzar: (i) Desarrollo de la sociedad de agentes; (ii) Simulación del comportamiento de la organización en un caso concreto de coordinación y adaptación de sus agentes; y (iii) validación del modelo de planificación propuesto mediante la simulación de la organización del caso de estudio.

---

## 1.4 ESTRUCTURA DE LA MEMORIA

---

Este documento se ha estructurado en ocho capítulos y diversos apéndices, todos ellos organizados en cinco partes.

La **primera parte** de esta memoria corresponde al capítulo 1, en el cual se hace una introducción a este trabajo de investigación. Se describe la problemática en torno al desarrollo de sistemas multi-agente basados en organizaciones, así como de la necesidad de obtener una mayor adaptación de estos sistemas frente a cambios en el entorno. Se presentan los objetivos, la hipótesis y la motivación que han llevado al desarrollo del modelo. Finalmente, se detalla la metodología de investigación aplicada y se realiza una breve descripción de la estructura de esta memoria.

La **segunda parte** presenta el estado del arte y los diferentes enfoques existentes relacionados con este estudio. Integra los capítulos 2, 3 y 4 de esta memoria.

En el capítulo 2, se analiza el estado del arte de los sistemas multi-agente. Se explicará el paradigma que envuelve a los agentes y sistemas multi-agente así como el camino recorrido desde su origen y las tendencias que más auge están teniendo actualmente. Y dado que la teoría de agentes ha evolucionado siguiendo distintas tendencias, se hará hincapié en agentes que utilizan el modelo deliberativo BDI y que son capaces de cooperar con otros agentes dentro de una "sociedad". Así como en conceptos como adaptación, coordinación y planificación de agentes.

El capítulo 3 continuará con el estado del arte de los sistemas multi-agente pero en este caso centrándose ya en las capacidades organizativas de los mismos. Se realiza un estudio de los modelos sociales de agentes así como las capacidades de coordinación y adaptación de los mismos.

El capítulo 4, titulado tecnologías complementarias, se centrará en aspectos importantes para el desarrollo posterior como son los sistemas de razonamiento basados en casos y los agentes virtuales inteligentes.

En la **tercera parte** de la memoria se presenta en detalle el modelo para organizaciones virtuales propuesto. Se definen las principales funcionalidades que desempeña y los componentes que lo integran, desde los tipos de agentes hasta el tipo de planificación empleada, así como las tecnologías base para explotar su potencial.

La **cuarta parte** de la memoria presenta la evaluación del modelo mediante un caso de estudio y los resultados experimentales y conclusiones obtenidos. Está formada por los capítulos 6 y 7. En el capítulo 6 se plantea un problema real enmarcado en un entorno dinámico, que puede servir como marco de pruebas para la aplicación del modelo. El problema trata de la automatización de algunos aspectos de la gestión de un sistema de guiado turístico. Entre estos aspectos destaca un sistema de planificación para los turistas en tiempo de ejecución y la asignación de guías turísticos a grupos de turistas. Todos ellos formarán una organización virtual simulada en la que se pondrá a prueba del modelo de organización planteado en capítulo anterior. En el capítulo 7 se presentan los resultados obtenidos con el análisis, diseño e implementación de la organización, planteando un escenario experimental para mostrar los resultados del modelo. Además se presentan las conclusiones obtenidas, se resumen las contribuciones de este trabajo al campo de los agentes inteligentes y se indican algunas de las posibles líneas de investigación futura.

La **última parte** del documento, la parte 5 "*Model Overview*", la forma un resumen en inglés de todo el trabajo realizado, tanto estudios iniciales, como propuesta, resultados y conclusiones obtenidas.

Finalmente se proporciona una lista de las **referencias bibliográficas** utilizadas para llevar a cabo este trabajo.

Los **anexos** permiten completar y detallar diferentes aspectos de la memoria que necesitan ser estudiados con mayor nivel de detalle. El anexo A describe el proceso de análisis y diseño seguido para el caso de estudio del sistema turístico, utilizando la guía metodológica para organizaciones de agentes GORMAS. En él se detalla tanto el análisis de la organización como su diseño a nivel estructural y dinámico. El anexo B presenta un resumen de las herramientas y plataformas utilizadas. Finalmente, el anexo C se presenta un resumen de los proyectos y publicaciones más destacados relacionados de alguna manera con el desarrollo de esta tesis.



## PARTE II. ESTADO DEL ARTE Y ENFOQUES



## 2 AGENTES Y SISTEMAS MULTI-AGENTE

---

El concepto de agente ya ha sido ampliamente estudiado en nuestros días. Sin embargo, existe un pequeño vacío en la consideración de agente como entidad que opera dentro de una sociedad. Y se trata de un tema especialmente importante para un agente, pues una de sus características básicas es que se encuentra situado en un entorno cooperando con otros agentes.

El presente capítulo estudia el concepto de agencia desde el punto de vista de su utilización en entornos cooperativos. Comienza con una revisión de la noción de agente inteligente. A continuación se expone una descripción de los sistemas multi-agente, junto con la definición de sociedad de agentes. El objeto es poner de manifiesto las tendencias actuales del desarrollo de SMA desde el punto de vista organizativo.

---

### 2.1 EL PARADIGMA DE AGENTES

---

Los retos a los que se enfrentan en nuestros días los desarrolladores de aplicaciones y sistemas de información son cada vez más complejos. Los agentes inteligentes son un nuevo paradigma para el desarrollo de aplicaciones software y actualmente son objeto de gran interés por parte de varios campos de la informática y de la Inteligencia Artificial (IA) [Jennings y Wooldridge, 1998].

La evolución del software, y más concretamente del que incorpora elementos de la inteligencia artificial, tiende a la creación de entidades con comportamientos y conductas similares a las de los humanos. La teoría de agentes se sustenta sobre el concepto de agente [Russel y Norvig, 1995]. Un agente es una entidad autónoma dotada con ciertas capacidades propias de los humanos. Puede verse como una evolución del concepto de objeto software, perfeccionada gracias a la influencia de la inteligencia artificial, que permite incorporar características como la racionalidad, la inteligencia, la autonomía o el aprendizaje. Al igual que los humanos, los agentes deben tener habilidades sociales y ser capaces de realizar trabajos o resolver problemas de forma distribuida [D'Inverno y Luck, 2004]. Se habla entonces de un sistema multi-agente, en el que los agentes cooperan e interactúan para conseguir los objetivos finales del sistema

---

#### 2.1.1 ¿Qué es un agente?

---

Debido al entorno multidisciplinar en el que se ha encontrado siempre el concepto de agente (abarca disciplinas como la informática, la Inteligencia Artificial, la Ingeniería del Software, las Bases de Datos, los Sistemas

Distribuidos, la Psicología o la Sociología) siempre ha sido patente la dificultad por dar una definición uniforme. Una de las definiciones más aceptadas es la siguiente [Wooldridge y Jennings, 1995]:

**Definición:** *“Un agente es un sistema computacional encapsulado que está situado en algún entorno, y que es capaz de actuar de forma autónoma y flexible en ese entorno para poder alcanzar sus objetivos de diseño”.*

Algunos de los conceptos introducidos en esta definición merecen una explicación más detallada. Por "sistema computacional encapsulado" se entiende que hay una distinción clara entre el agente y su entorno. Además, la definición implica que hay una frontera bien definida y una interfaz concreta entre el agente y su entorno. El aspecto clave de la definición es la autonomía, que se refiere al principio de que los agentes pueden operar por su cuenta, sin la necesidad de un guía humano. Un agente autónomo tiene el control sobre sus propias acciones y estado interno, es decir, un agente puede decidir si desea realizar una acción solicitada. La definición sitúa a un agente en un entorno particular, en que el agente puede "sentir y accionar". Esto nos lleva al comportamiento de respuesta. Además, la definición implica que los agentes pretenden alcanzar soluciones a problemas con límites bien definidos, para alcanzar un propósito específico, es decir, con objetivos específicos a lograr, y que exhiben un comportamiento activo flexible y pro-activo.

Para [Wooldridge y Jennings, 1995] el término agente disfruta de las siguientes propiedades:

- ❖ *Autonomía: Los agentes operan sin la intervención directa de los humanos u otros, y tienen alguna clase de control sobre sus acciones y estados internos*
- ❖ *Habilidades sociales: Los agentes interactúan con otros agentes (y posiblemente con humanos) por medio de algún lenguaje de comunicación de agentes.*
- ❖ *Reactividad: Los agentes perciben su entorno, (que puede ser el mundo físico, un usuario vía un interfaz gráfico, una colección de otros agentes, Internet, o quizás una combinación de todos ellos), y responde en un periodo determinado de tiempo a los cambios que en él se producen.*
- ❖ *Pro-actividad: Los agentes no se limitan a actuar en respuesta a su entorno, sino que son capaces de manifestar comportamientos dirigidos por metas tomando iniciativas propias”.*

Una definición más formal sería la siguiente: un agente es una entidad física o abstracta que puede percibir su ambiente a través de sensores, es capaz de evaluar tales percepciones y tomar decisiones por medio de mecanismos de razonamiento sencillos o complejos, comunicarse con otros agentes para obtener información y actuar sobre el medio en el que se desenvuelve a través



de ejecutores [Labidi y Lejouad, 1993]. En concreto, los agentes inteligentes se consideran una serie de entidades que tratan de imitar procesos de razonamiento o comportamientos humanos. Los agentes pueden ser empleados en diferentes áreas para facilitar tareas a los usuarios tales como recolección supervisión, filtrado de información, etc.

Existen otras definiciones para el término agente. Una de las más aceptadas es la que proporcionan [Russel y Norving 1995]:

**Definición:** *“La noción de un agente aparece como una herramienta para analizar sistemas, no una caracterización absoluta que divida el mundo en agentes y no agentes”.*

Para Russel, un agente se puede ver como “algo” capaz de percibir su entorno a través de sensores y de actuar en ese entorno a través de actuadores. De esta forma considera que un agente humano dispone de órganos sensores tales como ojos, oídos, etc., y que también dispone de órganos actuadores tales como manos piernas, boca, etc. De la misma forma, un agente robótico dispone de sensores de presión, de temperatura, cámaras, etc. y de motores como actuadores. Un agente software recibe claves, ficheros, paquetes de red como datos de entrada y actúa sobre su entorno mostrando cosas en pantallas, escribiendo ficheros o enviando paquetes de red.

A pesar de las diferencias encontradas en las definiciones que los autores pueden dar al término agente, encontramos una serie de características que éstos deben cumplir, las cuales se describen a continuación:

- ❖ Autonomía. Actuar sin la necesidad de intervenciones externas, ya sean humanos u otros agentes, y tener alguna clase de control sobre sus acciones y su estado interno.
- ❖ Situación. Situarse dentro de un entorno, ya sea real o virtual.
- ❖ Reactividad. Percibir su entorno y actuar sobre éste con la capacidad de adaptarse a sus necesidades.
- ❖ Pro-Actividad o Racionalidad. Tomar la iniciativa para definir metas y planes que les permitan alcanzar sus objetivos.
- ❖ Habilidad social. Interactuar con otros agentes, incluso con humanos.
- ❖ Inteligencia: Rodearse de conocimiento (creencias, deseos, intenciones y metas).

- ❖ Organización. Organizarse dentro de sociedades que siguen unas estructuras similares a las definidas en sociedades humanas o ecológicas.
- ❖ Aprendizaje. Habilidad de adaptarse progresivamente a cambios en entornos dinámicos, mediante técnicas de aprendizaje.

---

### 2.1.2 Clasificación de los agentes

---

Las numerosas definiciones y características relacionadas en la sección 2.1.1 sobre el concepto de agente provocan que la labor de clasificación de los diversos tipos de agentes no resulte fácil. Distintos autores [Russell y Norving, 1995], [Nwana, 1995], [Brenner *et al.*, 1998], [Franklin y Graesser, 1997], [Cetkovic y Parmee, 2002] han identificado clases de agentes estableciendo criterios que se basan en características comunes de los agentes o de los entornos de aplicación.

Una primera clasificación puede basarse en el tipo de programa utilizado para implementar las funcionalidades del agente, estableciendo el estado intermedio entre percepciones y acciones [Russell y Norving, 1995]. El criterio que utilizan para diferenciar los tipos de agentes es el tipo de programa utilizado para implementar el agente. De esta forma distinguen entre : (i) *agentes de reflejo simple*, agentes que reaccionan ante estímulos, proporcionando una respuesta y sin capacidad de memoria; (ii) *agentes reflejo con estado interno* (Reflex Agent with State) que incorporan un estado de memoria interno; (iii) *agentes basados en metas* (Goal Based Agent) que incorporan metas que dotan a este tipo de agente de capacidades deliberativas; y (iv) *agentes basados en utilidad* (Utility Based Agents) que incorporan el concepto de utilidad, una función que a cada estado le asocia su grado de utilidad y que permite la toma de decisiones racionales cuando satisfacer algunas metas implica un conflicto o cuando el agente puede desear obtener varias metas y no existe la certeza de lograr ninguna de ellas.

Otra clasificación podría hacerse atendiendo a la utilización de las teorías de agentes en el desarrollo de productos software comerciales [Brenner *et al.*, 1998]. Esta propuesta de clasificación llama agentes software a las entidades que presentan las características propias de los agentes y los clasifica atendiendo a su utilización en el desarrollo de software de carácter comercial: agentes de interfaz o asistentes personales, agentes de Internet, etc.

[Franklin y Graesser, 1997] proponen una clasificación para los agentes que se asimile a la clasificación realizada en el mundo animal. Para Franklin resulta razonable realizar una taxonomía para los agentes, ya que se trata de un software al que se le incorporan capacidades propias de las entidades de la naturaleza, y más concretamente de los humanos.

Una clasificación atendiendo al diseño conceptual es presentada por [Cetkovic y Parmee, 2002]. Teniendo en cuenta el diseño conceptual de los agentes identifican los siguientes tipos de agentes: (i) agentes interfaz, que ayudan al diseñador a interactuar con el sistema y a descubrir los detalles de bajo nivel que, en ocasiones, permanecen ocultos; (ii) agentes de búsqueda, que se ocupan de gestionar los procesos de optimización, cooperación, monitorización de la población, restricciones, etc; (iii) agentes de información, que trabajan con la información obtenida, buscando soluciones de interés.

A pesar de la diversidad, quizá una de las clasificaciones más completas sea la proporcionada por Nwana en [Nwana, 1995]. Esta clasificación se basa en varias dimensiones:

- ❖ Respecto a su *movilidad*, los agentes se dividen en móviles o estáticos, según su capacidad o no de desplazarse en una red. Los agentes estáticos son aquellos que solamente pueden ejecutarse en la máquina donde fueron iniciados. En contraste, un agente móvil es aquel que no se ve limitado al sistema donde comenzó su ejecución, sino que es capaz de transportarse de una máquina a otra a través de la red. Los agentes móviles se emplean, por ejemplo, para situaciones en las que no se puede tener una alta velocidad de comunicación a través de una red y el agente necesita acceder a datos que se encuentran en otro punto de esa red. En ese caso es rentable que el agente se desplace a ese punto y una vez tenga los datos (o en su caso, ejecutado el proceso), vuelva al sistema original. De este modo, el tránsito necesario a través de la red disminuye.
- ❖ En segundo lugar, un agente puede clasificarse como *deliberativo* o *reactivo*. Los primeros derivan del paradigma del pensamiento deliberativo: los agentes poseen un modelo simbólico interno de razonamiento y se ocupan de la planificación y la negociación con el objetivo de conseguir coordinarse con otros agentes. Los agentes reactivos, por el contrario, no poseen ningún modelo simbólico interno de su entorno y actúan empleando un comportamiento del tipo estímulo respuesta, respondiendo al estado actual del entorno en que están embebidos.

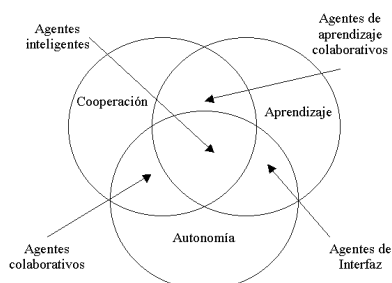


Figura 1. Clasificación de los agentes según su característica primaria [Nwana, 1995].

- ❖ En tercer lugar, los agentes pueden clasificarse según los *atributos primarios* que deberían mostrar. Nwana distingue tres principales: autonomía, aprendizaje y cooperación. De este modo existen los agentes *autónomos*, *de aprendizaje* y *cooperativos*. Las combinaciones dos a dos de estas características originan los agentes *colaborativos* (cooperación y autonomía), *colaborativos de aprendizaje* (aprendizaje y cooperación) y de *interfaz* (aprendizaje y autonomía). Finalmente, la combinación de los tres elementos da como resultado los agentes inteligentes. Esta distribución se muestra en la Figura 1.
- ❖ En algunas ocasiones, los agentes pueden ser clasificados por el *papel que desempeñan*. Por ejemplo, los agentes *de información* o los agentes *de Internet*. Estos tipos de agentes se dedican a la búsqueda y procesamiento de información en una red, como en el caso de Internet.
- ❖ Por último, Nwana habla de agentes *híbridos* para referirse a los que combinan dos o más de las categorías anteriores.

A modo de conclusión, se ha podido observar cómo prácticamente todas las clasificaciones se fijan en los atributos de los agentes como criterio principal. Existen una serie atributos que pueden servir de referencia, como son la autonomía, la pro-actividad, el aprendizaje, y la cooperación. Las clasificaciones convergen especialmente en la necesidad de diferenciar aquellos agentes que son simplemente reactivos de aquellos que incorporan algún mecanismo de razonamiento. Además se hace hincapié en la existencia de agentes híbridos, que flexibilizan las clasificaciones y permiten la integración de más de un criterio de clasificación.

---

### 2.1.3 Arquitecturas

---

La forma en que se descomponen los agentes en un conjunto de módulos y cómo estos módulos interactúan entre sí para alcanzar la funcionalidad requerida, viene dada por la arquitectura de agente seleccionada [Mas, 2005].

Se han propuesto varias arquitecturas, las cuales clasifican a los agentes que viven en ellas [Wooldridge, 1999]:

- ❖ Agentes basados en lógica: el razonamiento y la toma de decisiones se realiza a través de lógica y deducción [Genesereth y Nilsson, 1987][Lesperance et al., 1996][Fischer, 1994].
- ❖ Agentes reactivos: la toma de decisiones es llevada a cabo como un mapeo directo de una situación a una acción [Brooks, 1986][Maes, 1990].
- ❖ Agentes BDI (Belief-Desire-Intention, Creencia-Deseo-Intención): la toma de decisiones depende de la manipulación de la representación de de las creencias, deseos e intenciones del agente [Bratman et al., 1988], [Rao, Georgeff, 1992].
- ❖ Agentes basados en capas: la toma de decisiones se realiza a través de varias capas de software, cada una llevando el razonamiento sobre el entorno a diferentes niveles de abstracción [Ferguson, 1995].

La planificación en la toma de decisiones es un criterio clave a la hora de conocer el funcionamiento de los agentes. Los sistemas de planificación utilizan modelos de representación del conocimiento y razonamiento de tipo simbólico y su modo de actuación está definido por la necesidad de satisfacer unos objetivos básicos, para la elaboración de un plan. Una desventaja importante de estos sistemas es su aplicación en entornos de tiempo real, ya que estos algoritmos de planificación algunas veces no pueden responder en el tiempo necesario que exige el sistema. Debido a esto se buscan nuevas alternativas que implementen nuevos modelos de representación y conocimiento. La siguiente clasificación atiende a tres tipos de arquitecturas que se diferencian respecto al modelo de razonamiento:

- Arquitectura Deliberativa: las arquitecturas deliberativas emplean modelos de representación simbólica del conocimiento y suelen estar basadas en la teoría clásica de planificación partiendo de un estado inicial en el que existen una serie de planes y un estado final al que se quiere llegar. Estos agentes están dotados de un sistema de planificación que permite determinar los pasos que se han de efectuar para alcanzar los objetivos. Los agentes intencionales (BDI) se pueden implementar utilizando una arquitectura deliberativa. Estos agentes están basados en una serie de creencias e intenciones que utilizan para generar planes [Jennings, 1993]. Entre las arquitecturas intencionales la más ampliamente extendida y estudiada es la que se basa en el modelo BDI [Rao, Georgeff, 1992].

- **Arquitectura Reactiva:** carecen de razonamiento simbólico complejo y de conocimiento o representación de su entorno, por lo que sus mecanismos de comunicación con otros agentes son muy básicos. Los agentes que utilizan este tipo de arquitectura reciben estímulos de su entorno y reaccionan ante ellos modificando sus comportamientos y el mismo entorno [Maes, 1990].
- **Arquitectura Híbrida:** son arquitecturas intermedias entre las dos anteriores. Los agentes de este tipo incluyen comportamientos reactivos y deliberativos, generando un ciclo percepción-decisión-acción. El comportamiento reactivo se utiliza para reaccionar ante eventos que no requieran decisiones complejas sobre ciertas acciones.

De las arquitecturas descritas, hay que prestar especial atención a la arquitectura basada en modelo BDI. Por un lado, esta arquitectura se ha convertido en un estándar de facto para los modelos de agente y es aceptada por la FIPA<sup>1</sup>, y, por otro lado, es suficientemente genérica como para permitir el modelado de agentes, tanto naturales como artificiales.

A lo largo de esta tesis, se argumentará que los modelos de agente de las arquitecturas no pueden basarse solamente en las especificaciones internas de los agentes individuales. Es requisito indispensable la cooperación del individuo (agente) dentro de una comunidad (organización). Al ser una arquitectura genérica, BDI ofrece el mejor enfoque a este requisito.

---

#### 2.1.4 Agentes BDI

---

La Inteligencia Artificial ha estudiado lo que denomina “actitudes mentales” en los agentes inteligentes, y que definen su capacidad de razonamiento. Estas actitudes están relacionadas con la observación y detección de capacidades ligadas con estados mentales humanos (conocimiento, creencias, deseos y metas). Estos estados pueden representarse por medio de actitudes, que pueden ser informativas, de motivación o sociales.

- Las actitudes informativas consideran lo que se debe hacer con la información a la cual tienen acceso los agentes, incluyendo la relativa a su entorno. Entre estas actitudes destacan el conocimiento (por definición verdadero) y las creencias.
- Las actitudes de motivación son las directamente relacionadas con las posibles acciones que un agente pueda llegar a tomar. Como ejemplo

---

<sup>1</sup> FIPA (Foundation for Intelligent Physical Agents) <http://www.fipa.org/> [Último acceso 19/08/2009]

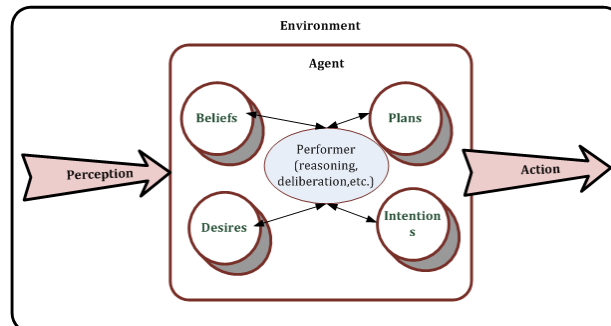
tenemos las actitudes de elecciones, planes, metas, deseos, preferencias, etc.

- Por último, las actitudes sociales son externas y están relacionadas con los permisos y obligaciones, ya sean de carácter político, moral o racional.

El modelo BDI [Bratman et al., 1988] [Bratman, 1987] limita estas actitudes mentales de los agentes a tres: creencias (*Beliefs*), deseos (*Desires*) e intenciones (*Intentions*).

- ❖ *Creencias*. Conjunto de proposiciones que el agente acepta como verdaderas. Dicho de otra forma, lo que el agente conoce de su entorno (su modelo del mundo y del resto de los agentes). Obsérvese que aunque el agente lo tome como verdadera, esa creencia puede resultar falsa, a diferencia del conocimiento, que por definición es siempre verdadero.
- ❖ *Deseos*. También llamados frecuentemente metas u objetivos. Denotan una propiedad o conjunto de propiedades del entorno que el agente quiere que sean verdaderas, pero que no se encuentran actualmente entre sus creencias. Es decir, representa algún estado final deseado.
- ❖ *Intenciones*. Conjunto de acciones planificadas por el agente que le permiten llegar a un estado deseado. Estos planes pueden ser interrumpidos en un momento dado y son realimentados con los posibles cambios que pudieran producirse en el entorno.

Además de las creencias, los deseos y las intenciones, es común definir un conjunto de planes, es decir, una secuencia de acciones [Corchado et al. 2008]. Si un agente tiene la intención de alcanzar un objetivo, debe tener la intención de ejecutar el plan que permita alcanzarlo, y además, debe creer que ese plan logrará dicho objetivo [Cavedon y Rao, 1996]. Por lo tanto, la arquitectura básica de un agente BDI, mostrada en la Figura 2, es un conjunto de creencias, planes, deseos e intenciones, procesadas por un intérprete que puede desempeñar diversas funciones, entre ellas el filtrado, el razonamiento y la deliberación.



**Figura 2. Arquitectura básica de un agente BDI**

Posiblemente el modelo BDI ha sido el más difundido y el más estudiado dentro de los modelos de razonamiento de agentes. Hay varias razones para que esto ocurriera pero quizás la más convincente es que el modelo BDI combina elementos interesantes: un apreciable modelo filosófico de razonamiento humano fácil de comprender, un número considerable de implementaciones [Georgeff y Lansky, 1987], como por ejemplo sistemas de control de procesos, procesos de decisión en negocios, etc., y se ha desarrollado una semántica lógica abstracta y elegante, la cual ha sido aceptada por la comunidad científica [Georgeff et al., 1999][Rao y Georgeff, 1998].

En este modelo, las entradas al sistema son eventos recibidos mediante una cola de eventos. El sistema puede reconocer tanto eventos internos como externos. Se asume que los eventos son indivisibles y se reconocen cuando se completan, no cuando ocurren. Por otro lado, la salida del sistema son acciones (también indivisibles) realizadas por una función de ejecución. Basándose en el estado actual del sistema y de los eventos presentes en la cola, se seleccionan las opciones a ejecutar. La implementación de este modelo suele llevar los siguientes pasos, repetidos de forma continua:

- 1) Al inicio de cada ciclo, un generador de opciones lee la cola de eventos y genera una lista de opciones.
- 2) Un liberador de opciones selecciona un subconjunto de opciones y las agrega a la estructura de intenciones.
- 3) Si existe una intención que realiza una acción en este punto, el agente la ejecuta.
- 4) Si algún evento externo ha ocurrido durante el ciclo, es agregado a la cola de eventos.
- 5) Los eventos internos son agregados.



- 6) El agente modifica la estructura de deseos e intenciones eliminando aquéllos alcanzados con éxito así como aquellos deseos imposibles y acciones no realizables.

Los agentes deliberativos BDI se modelan utilizando una estructura abstracta, basada en la lógica de situaciones (o mundos posibles), denominada árbol temporal con múltiples futuros y un solo pasado[Rao y Georgeff, 1991]. En este árbol, cada nodo es una situación, y las ramas son las opciones del agente en un momento dado. Para cada situación se definen una serie de nuevas situaciones que el agente puede alcanzar desde el punto de vista de las creencias (situaciones que se consideran posibles), de los deseos (situaciones que se desean alcanzar) y de las intenciones (situaciones que se intentan alcanzar). Para este modelo es necesario que existan ciertas relaciones entre las creencias, los deseos y las intenciones del agente:

- ❖ Compatibilidad entre creencias y objetivos. Si el agente desea alcanzar un objetivo, debe creer que dicho objetivo es cierto.
- ❖ Compatibilidad entre objetivos e intenciones. Antes de que el agente se enfoque en una intención, éste debe haberla formulado como deseo.
- ❖ Las intenciones conducen a acciones. Es necesario que el agente ejecute las intenciones a través de acciones simples.
- ❖ Relación entre creencias e intenciones. El agente debe creer en sus propias intenciones.
- ❖ Relación entre creencias y objetivos. El agente debe conocer sus objetivos y deseos.
- ❖ No hay retrasos infinitos. Cuando un agente adopta una intención para alcanzar un objetivo, debe continuar hasta un determinado momento finito.

Los agentes deliberativos BDI han sido exitosamente utilizados en diversos sistemas desarrollados por el grupo de investigación BISITE<sup>2</sup> [Corchado et al., 2003][Corchado et al., 2005][Corchado et al., 2006][Bajo et al., 2006]. Estos agentes proporcionan soluciones en entornos dinámicos e inciertos. Además, son capaces de enfrentarse a problemas del mundo real, incluso cuando sólo cuentan con una visión parcial del problema y con un número limitado de recursos. Los agentes deliberativos BDI son capaces de proporcionar altos niveles de adaptabilidad, tanto a las condiciones del entorno como a las necesidades de los usuarios. Esto resulta importante de cara a individualizar las

---

<sup>2</sup> Biomedicina, Sistemas Informáticos Inteligentes y Tecnología Educativa <http://bisite.usal.es> [Última acceso 22/11/2009]

acciones que puedan ser tomadas en cuenta para afrontar distintas situaciones de forma dinámica y autónoma.

La utilización de agentes deliberativos BDI es una pieza fundamental en el desarrollo del modelo propuesto, el cual se describe en detalle en el capítulo 5 de este documento.

---

### 2.1.5 *Planificación y agentes*

---

Respecto a las características esenciales de un agente, una cosa es clara: siempre existirá un objetivo a alcanzar y unas acciones que el agente pretenderá llevar a cabo para conseguirlo. Es un hecho aceptado dentro de la comunidad de investigadores en Inteligencia Artificial, que en la estructura de un agente inteligente, para que pueda ejecutar las acciones que le permitan alcanzar sus objetivos, debe existir un componente imprescindible: un motor planificador. [Georgeff y Lansky, 1987] incluso afirman que la planificación es el componente central de la actividad de un agente.

Un planificador en Inteligencia Artificial se define de forma general como un sistema que posee un objetivo, una representación del estado actual del mundo, y genera una secuencia de acciones. Para un agente es exactamente lo mismo: un agente posee un estado que pretende alcanzar, un conocimiento del sistema, y se ejecutan planes de acción para alcanzar un estado deseado. Un agente, en la manera en que interacciona con el entorno, es esencialmente un planificador, por lo que es posible desarrollar un método formal para el modelado y razonamiento de agentes. A continuación se presentan las características planificadoras de las principales arquitecturas de agentes, mostrando primero los principales tipos de planificación identificados dentro de la Inteligencia Artificial.

#### 2.1.5.1 *Planificación en IA*

La planificación ha sido estudiada desde muchos puntos de vista o perspectivas dentro de la Inteligencia Artificial [Ghallab y Laruelle, 1994][Castillo, 1998]. Los planes en la IA son vistos como una secuencia de acciones de un agente, de tal manera que es capaz de cambiar su entorno y lograr uno o más objetivos. La representación del entorno, de las acciones y de los cambios es una parte muy importante de la teoría de la planificación. Se han propuesto numerosos lenguajes que permiten obtener estas representaciones. En general se suele representar un árbol de búsqueda y se suele distinguir entre la representación de los nodos del árbol como estados [Fikes y Nilsson, 1971] [Veloso y Rizzo, 1998] o como planes parciales [Sacerdoti, 1975] [McAllester y Rosenblitt, 1991] [Penberthy y Weld, 1992].

Es frecuente encontrar el término de planificación ligado a la toma de decisiones en ámbitos relacionados con el control, en el que las decisiones a tomar se centran en la asignación de recursos y la optimización de ciertas

funciones, derivándose la complejidad de factores tales como restricciones, costes, etc. Un típico problema de optimización busca la óptima distribución de recursos que cumpla con las restricciones impuestas, a partir de un *conjunto de tareas ya establecido en un dominio concreto*. Por poner un ejemplo, en control de la producción: [Dominguez et al., 1995] [Hernández, 2001]. La formalización y estudio de este tipo de problemas escapa al ámbito de la investigación llevada a cabo es esta tesis. La noción de "Planificación" empleada en este documento, hace referencia a su acepción más abstracta, abordando la *construcción de un modelo propio que permita la generación de un plan*. Así, un modelo de planificación contiene una representación del mundo y de las acciones disponibles. Cuando se genera un plan, estas acciones pueden interactuar de diferentes maneras, para lo que existen una serie de mecanismos más o menos sofisticados de resolución.

Los modelos de planificación dentro de la IA tradicionalmente se han dividido en dos tipos: planificación clásica y planificación moderna.

La **planificación clásica** es el primer tipo de planificación que se conoce. Según [Yang,1997], su definición es la siguiente:

**Definición:** *Se entiende por planificación clásica la generación de planes para alcanzar unas especificaciones objetivo a partir de unas especificaciones iniciales, definidas ambas completamente, en situaciones en las que las particularidades más relevantes del mundo son conocidas y en las que el éxito del plan no depende de los cambios en el entorno (no existen, por tanto, eventos exógenos).*

Se trata de modelos que imitan el pensamiento humano [Fikes et al., 1972], [Newell y Simon, 1963]. La planificación clásica necesita una descripción del mundo que contenga todas las posibles configuraciones, un estado inicial, un estado meta final y un conjunto de operadores o acciones que permitan cambiar el estado del mundo (manipular los estados). La formulación más intuitiva de un problema de planificación clásica tiene 3 entradas: una descripción del estado inicial del mundo, una descripción del estado del mundo objetivo y una descripción de las posibles acciones que se pueden realizar. Más formalmente, un problema de planificación clásica  $P$  se representa como una tupla  $(D, I, O)$  en la que  $D$  representa una *Teoría del Dominio* (las acciones que se pueden realizar en el espacio del problema, restricciones generales del espacio del problemas, etc.),  $I$  son las especificaciones iniciales y  $O$  representa las especificaciones objetivo. A partir de  $P(D, I, O)$  el problema consiste en encontrar un plan como combinaciones de acciones, que permitan pasar del estado del mundo  $I$  a un estado del mundo que contenga las especificaciones objetivo [Weld, 1999]. De esta forma un plan se puede representar tal y como se muestra en la Figura 3 como la secuencia de acciones que transforma el estado inicial de partida en el estado meta final.

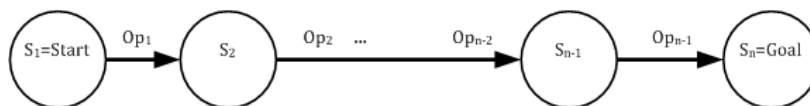


Figura 3. Representación de plan como secuencia de acciones en planificación clásica.

En este sentido, es necesario distinguir entre *planificación lineal* y *planificación no lineal*: en la *planificación lineal* existe una relación de linealidad entre la forma de ir resolviendo los subobjetivos y su posterior ejecución. El conformado del plan se realiza en el mismo orden de su ejecución, alcanzándose objetivos parciales para, a partir de este punto, abordar nuevos subobjetivos. El resultado de la planificación lineal es una secuencia totalmente ordenada de pasos, normalmente ejecutable por un único agente. En los *planificadores no lineales*, no existe linealidad en el orden de resolución de subobjetivos y su orden de ejecución. Dentro de este colectivo destaca la *planificación parcialmente ordenada* (PPO), donde la búsqueda se desarrolla en un espacio de planes, a diferencia del espacio de estados en los planificadores lineales. Cada nodo generado, posible plan solución, se somete a críticas (término original del planificador NOAH [Sacerdoti, 1975], generalizable a toda rutina precompilada capaz de modificar un plan y corregir defectos) y cada nuevo nodo sucesor es un plan cada vez más refinado que finalmente converge a un plan solución.

Generalmente, la planificación clásica ha sido asociada a los modelos de planificación lineal mientras que la planificación moderna se asocia a modelos que no siguen los criterios de linealidad, introduciendo criterios de orden parcial.

La **planificación moderna** no se ajusta a las suposiciones realizadas por la planificación clásica. Se introducen técnicas como la teoría de grafos o la utilización de heurísticas, que permiten reducir considerablemente el espacio de búsqueda. También se tienen en cuenta criterios para satisfacer tanto el cumplimiento de restricciones como el consumo de recursos. A continuación se presentan las principales técnicas estudiadas:

- ❖ Planificación basada en grafos. Se trata de un tipo de planificación basada en la teoría de grafos de Euler (1736). Consiste en la creación de grafos en los que se representan todos los posibles planes a llevar a cabo antes de comenzar con el proceso de búsqueda. Un grafo de planificación es una estructura compacta que codifica un problema de planificación y almacena, explícitamente, muchas de las restricciones entre acciones y proposiciones. Se trata de un grafo similar a los utilizados en programación dinámica [Bellman y Dreyfus, 2001], modelando la parte estática y dinámica de un plan mediante nodos y arcos respectivamente. En general se basan en generadores de planes

como STRIPS (*Stanford Research Institute Problem Solver*) o PDDL (*Planning Domain Definition Language*). Un ejemplo es GRAPHPLAN<sup>3</sup> [Blum y Furst, 1997], que utiliza un grafo para representar el mundo como proposiciones (niveles pares del grafo) y acciones (niveles impares), lo que permite representar, además, varios tipos de conflictos y razonar búsquedas simplificaciones. El algoritmo presenta 2 fases: (i) la expansión del grafo y (ii), la extracción de soluciones. La fase de expansión extiende el grafo hacia delante en el "tiempo" hasta que se ha satisfecho alguna condición necesaria (pero insuficiente, en el caso general) para la existencia de un plan. Comienza entonces una segunda fase, una búsqueda hacia atrás sobre el grafo tratando de encontrar un plan solución que resuelva el problema. En caso de no ser posible, se repite el ciclo expandiendo iterativamente el grafo. Típicamente, la fase de extracción de la solución se modela como un CSP (Problema de Satisfacción de Restricciones) [Sam-Haroud y Faltings, 1996].

- ❖ Planificación basada en satisfactibilidad. En este tipo de planificación, un problema se modela como un conjunto de axiomas con la propiedad de que algún modelo de los axiomas se corresponda con un plan válido. Se utilizan métodos automáticos para codificar problemas de planificación como fórmulas proposicionales que se resuelven automáticamente mediante algoritmos estocásticos [Kautz y Selman, 1996], [Weld, 1999]. Un ejemplo de este tipo de planificación es SATplan4 [Kautz et al., 2006].
- ❖ Razonamiento basado en modelos. El dominio de planificación se define mediante un modelo semántico, donde las propiedades deseadas se describen mediante formulas lógicas. Algunos ejemplos son UMOP5 (*Universal Multi-agent Obdd-based Planner*) [Jensen y Veloso, 2000], CMBP (*Conformant Model Based Planner*) [Cimatti y Roveri, 2000]. Tanto UMOP como CMBP6 utilizan diagramas de decisión binarios, *Binary Decision Diagrams* (BDDs), que son estructuras de datos simbólicos para representar y manipular de forma eficiente conjuntos de estados y acciones.
- ❖ Heurística
  - Planificación heurística. La planificación heurística suele utilizarse en problemas dependientes del dominio, en los que el planificador usa su conocimiento del dominio en el que se encuentra para aplicar heurísticas específicas que le permitan controlar su operación. Se basan en la realización de una búsqueda en el espacio de estados por medio de la utilización de una heurística

---

<sup>3</sup> Versión original de GRAPHPLAN en C:

<http://www.cs.cmu.edu/afs/cs.cmu.edu/user/avrim/www/graphplan.html> [Último acceso 27/12/2009]

<sup>4</sup> <http://www.cs.rochester.edu/~kautz/satplan/index.htm> [Último acceso 27/12/2009]

<sup>5</sup> <http://www.cs.cmu.edu/~runej/systems/umop12.html> [Último acceso 27/12/2009]

<sup>6</sup> <http://sra.itc.it/people/cimatti/cmbp/> [Último acceso 27/12/2009]

que debe poder ser extraída automáticamente de la codificación del problema y combinada con algoritmos de búsqueda estándar. El éxito depende de la elección de la heurística, del algoritmo de búsqueda y de la dirección de esa búsqueda. Ejemplos de este tipo de planificación son GRT (*Domain Independent Heuristic for STRIPS Worlds based on Greedy Regression Tables*) [Refanidis y Valvas, 2001], HSP (*Heuristic Search Planner*) [Bonet y Geffner, 1999]. GRT es un planificador independiente del dominio que utiliza representación STRIPS.

- Redes de tareas jerárquicas. La planificación jerárquica consiste en la resolución de un problema de forma abstracta, es decir, de la construcción de un plan abstracto sin especificar los detalles concretos. Posteriormente el plan abstracto se refina hasta conseguir un nivel detallado y se continúa con este proceso hasta que se consigue el plan final. Se realizan representaciones jerarquizadas de acciones. Los objetivos y las acciones de un problema se representan en una estructura jerarquizada de abstracción [Castillo, 1998]. La planificación jerárquica se distingue de la planificación clásica en que en vez de planificar un conjunto de metas, planifica un conjunto de acciones que llevan a cabo una actividad o tarea y para ello va descomponiendo tareas en subtareas de acuerdo con la definición de diversos métodos. Se diferencian:

- *Simple Task Networks* (STNs): la descomposición se aplica en cumplimiento de unas precondiciones según una jerarquía.
- *Hierarchical Task Networks* (HTNs): la descomposición se realiza en un conjunto de tareas que cumplen ciertas restricciones. En HTN existen tres tipos distintos de tareas: tareas objetivo que son propiedades del mundo circundante que se desean hacer ciertas, tareas primitivas que son tareas que se pueden realizar de forma directa mediante la ejecución de una acción simple, y tareas compuestas que son tareas que describen cambios en cuya consecución puede ser necesario coordinar diferentes tareas objetivo y tareas primitivas.

Este tipo de planificación es muy eficiente si se define bien el conocimiento, pero los dominios son más complicados de definir. En este tipo encontramos planificadores como ABSTRIPS [Sacerdoti, 1975] y ALPINE [Knoblock, 1993] que van estableciendo niveles de abstracción en las precondiciones de los operadores; NOAH [Sacerdoti, 1977] y MOLGEN [Stefik, 1981], que van refinando los operadores sucesivamente; O-PLAN [Currie and Tate, 1991], que va preprogramando en qué se divide cada operador.

- ❖ Planificación con tiempo y recursos. Si tenemos en cuenta una planificación en la que las actividades a planificar comparten recursos durante la ejecución, se hace necesario modelar permitiendo que ciertos componentes usen recursos durante un intervalo de tiempo en el que se ejecutan. En el contexto de procesamiento de restricciones se establecen restricciones con respecto a tres tipos de recursos: recursos binarios con capacidad sencilla, recursos renovables y recursos consumibles [Castillo, 1998]. Además, si se elimina la restricción de que las acciones sean instantáneas como ocurría en la planificación clásica, es necesario tener una representación de la noción de tiempo. La planificación temporal tiene en cuenta “*deadlines*” y duración de las acciones que ejecuta el agente de tal forma que se cumplan las acciones del plan dentro de las restricciones temporales concretas [Castillo, 1998].
- ❖ Planificación basada en casos. Es un tipo de planificación basada en la experiencia. Los elementos que componen un plan se obtienen a partir de planes o fragmentos de planes que se hayan llevado a cabo en el pasado. La planificación basada en casos permite utilizar experiencias pasadas para resolver nuevos problemas [De Paz et al., 2009]. La posibilidad de utilizar fragmentos de planes o sub-planes permite una gran flexibilidad a la hora de replanificar, factor fundamental por el que este tipo será utilizado a lo largo del desarrollo de este trabajo y por el que este tipo de planificación será vista de forma más detallada (sección 4.1.1).

#### 2.1.5.2 Características planificadoras de agentes

Teniendo en cuenta las propiedades de planificación, las principales arquitecturas de agentes muestran las siguientes características planificadoras:

- ❖ En las arquitecturas *deliberativas*, un agente deliberativo puede definirse tomando como propuesta la definición de [McCarthy, 1978] de sistema planificador clásico: aquel sistema que, a partir de una base de datos donde se almacenan formulas en algún lenguaje simbólico que representan su conocimiento, opera siguiendo el ciclo “Percepción - Deliberación - Acción”, con un mecanismo de razonamiento y decisión de las acciones que se van a ejecutar basado esencialmente en alguna forma de inferencia lógica. Este planteamiento presenta dos problemas: (i) cómo representar el mundo real de forma simbólica, en tiempo real, para que tal descripción del mundo sea útil, y (ii) cómo lograr que los agentes manipulen la información representada, en tiempo real, para que los resultados de las acciones sean útiles. En resumen, las arquitecturas deliberativas de planificación, trabajan en un espacio de decisiones desvinculado del mundo real, sobre el que posteriormente ejecutan la acción elegida. La ventaja de usar una arquitectura de este tipo es que permite describir todos los pasos del

- comportamiento de un agente racional con un alto grado de confianza en la responsabilidad del diseño. Sin embargo, muchos de los requisitos de los agentes entran en oposición con el ordenado mundo de la planificación clásica: (i) los agentes poseen creencias inconsistentes, (ii) tienen múltiples disposiciones que compiten, y (iii) deben ser sistemas robustos frente a entornos hostiles y desconocidos.
- ❖ En las arquitecturas no *deliberativas* se parte de la idea de que el mecanismo de control del sistema se debe descomponer de acuerdo con el comportamiento externo deseado [Rosenblatt y Payton, 1989]. Se evita la manipulación de representaciones explícitas. Las arquitecturas representativas de este enfoque, basado en comportamiento, son las arquitecturas de *Subsuncción* [Brooks, 1991] y las arquitecturas basadas en *argumentos en ejecución* (“*running arguments*”) [Agre y Chapman, 1990]. En ambas, la característica esencial es su reactividad. No hay representación explícita de objetivos o de otras estructuras simbólicas sino módulos reactivos que se acoplan para percibir y actuar.
  - ❖ En las arquitecturas *híbridas* se trata de aprovechar las características deseables tanto de las arquitecturas deliberativas como de las no-deliberativas (o reactivas). Los agentes con arquitecturas deliberativas tienen poca capacidad de respuesta al medio si los cambios en el entorno exigen respuestas rápidas para sobrevivir. En el caso de los agentes con arquitecturas no-deliberativas, no parece demasiado claro que sean capaces de comportarse con estrategias a largo plazo de manera eficaz. La metodología híbrida intenta aprovechar las ventajas de las arquitecturas deliberativas y no-deliberativas para el diseño de agentes, utilizándolas de forma complementaria.

Es importante en este trabajo comprender perfectamente los conceptos de planificación en los agentes, ya que en el modelo presentado se encuentra, como concepto fundamental, la utilización una planificación de agentes inmersos en entornos altamente cambiantes que proporciona la posibilidad de re-planificar en tiempo de ejecución. Dadas las características de los distintos tipos de planificadores, se opta por utilizar un sistema de planificación basada en casos.

La planificación basada en casos permite utilizar experiencias pasadas para resolver nuevos problemas. La posibilidad de utilizar fragmentos de planes o sub-planes permite una gran flexibilidad a la hora de replanificar. Se trata de definir un espacio de planificación en el que un agente represente planes conocidos en términos de sus objetivos acumulados y los recursos consumidos según una jerarquía de planes en término de una medida de la eficiencia de cada plan. En este espacio, se representan planes de mínimo riesgo en un entorno cambiante, que cumplen las condiciones de optimalidad de Bellman [Bellman, 1957], lo que permite trabajar con técnicas de programación dinámica. Para tratar con las restricciones que imponen los problemas CSP



[Sam-Haroud y Faltings, 1996], se construye un espacio de las restricciones (un hiperplano, en general no-euclídeo) en el espacio de planificación, sobre el que se aplican técnicas basadas en cálculo variacional [Schutz, 1993] para calcular los planes geodésicos sobre él. Como resultado se obtiene una familia de planes, compatibles con las restricciones, que constituyen el modo de pasar de un estado actual a un estado objetivo de un agente manteniendo condiciones de eficiencia constante a lo largo del proceso. Como veremos, la planificación no será abordada de una manera individual ya que los agentes estudiados pueden coordinar su conocimiento, objetivos, habilidades y planes juntamente para tomar una acción o resolver una meta global dentro de una organización.

---

## 2.2 SISTEMAS MULTI-AGENTE

---

La utilidad de cualquier tecnología, incluida la de los sistemas multi-agente (SMA) puede ser juzgada en dos direcciones: (i) su habilidad para resolver nuevos tipos de problemas y (ii) su habilidad para mejorar la eficiencia de las soluciones actuales [Jennings y Worlddridge, 1998]. Teniendo en cuenta esto, los agentes y sistemas multi-agentes proporcionan un modo natural de caracterizar sistemas inteligentes. Inteligencia e interacción son conceptos ineludiblemente unidos y la tecnología de agentes refleja esta condición muy bien.

Cuando hablamos de SMA, extendemos la idea de un agente en solitario completándola con una infraestructura para la interacción y comunicación. Los SMA idealmente tienen las siguientes características [Huhns y Stephens, 1999]:

- ❖ Son típicamente abiertos y tienen un diseño no centralizado.
- ❖ Contienen agentes autónomos, heterogéneos y distribuidos, con diferentes "personalidades" (cooperativo, egoísta honesto, etc.).
- ❖ Proporcionan una infraestructura para especificar comunicación y protocolos de interacción.

Las aplicaciones del paradigma de agentes pueden ser clasificadas en tres clases [Dignum, 2004]: sistemas abiertos, sistemas complejos y sistemas ubicuos.

- ❖ Los *sistemas abiertos* son sistemas en los cuales la estructura es capaz de cambiar dinámicamente. Sus componentes no son conocidos a priori, cambian en el tiempo y pueden ser heterogéneos. Un ejemplo de un sistema abierto es Internet: cualquier sistema informático que debe trabajar en internet tiene que ser capaz de operar con organizaciones de muy distinta índole y además sin guiado constante a los usuarios. Este tipo de funcionalidad requiere de técnicas de negociación y cooperación que encontramos también en el dominio de los SMA.

- ❖ Los *sistemas complejos* están relacionados con dominios grandes, impredecibles, en definitiva, complejos. Las herramientas más poderosas para hacer frente a la complejidad de los sistemas son la modularidad y la abstracción. Un problema a resolver con agentes puede dividirse en un número de sub-problemas de menor complejidad, que son más fáciles de manejar. Esta descomposición permite a los agentes emplear la solución posible más apropiada para resolver un sub-problema
- ❖ Los *sistemas ubicuos* tienen el objetivo de mejorar el uso de un sistema informático mediante la utilización de ordenadores disponibles en un entorno físico, normalmente distribuido, pero haciendo todo de manera invisible al usuario. Este tipo de sistemas son más o menos lo contrario de la realidad virtual. Donde la realidad virtual pone a la gente dentro de un mundo generado por ordenador, la computación ubicua obliga al ordenador a "vivir" en el mundo con la gente [Weiser, 1993]. El sistema tiene que cooperar con el usuario para alcanzar su objetivo. Las aplicaciones tienen que comportarse como un agente inteligente.

Los SMA abiertos deben permitir la participación de agentes heterogéneos, con arquitecturas e incluso lenguajes distintos [Zambonelli et al., 2003]. Por esta razón, no se puede confiar en el comportamiento de los agentes, siendo necesario establecer controles en base a normas o reglas sociales. Para ello, los desarrolladores se han centrado en los aspectos organizativos de la sociedad de agentes, guiando el proceso de desarrollo de los sistemas mediante los conceptos de organización, normas, roles, etc.

Actualmente, investigaciones centradas en el diseño de SMA desde el punto de vista organizacional están ganando terreno. Cada vez es más fuerte la idea de que modelar las interacciones de un SMA no puede estar relacionado solamente con el propio agente y sus capacidades de comunicación, si no que es necesario una ingeniería organizacional. Los conceptos de reglas [Zambonelli, 2002], normas e instituciones [Esteva et al., 2001] y estructuras sociales [Parunak, Odell, 2002] que veremos en este estudio, nacen de la idea de que es necesario un nivel de abstracción mayor, independiente del agente y que defina explícitamente la organización en la que los agentes viven.

Los agentes en un SMA basado en conceptos organizacionales se coordinan e intercambian servicios e información; son capaces de negociar y llegar a acuerdos; y pueden llevar a cabo otras acciones sociales más complejas. Coordinación y comunicación son temas muy importantes dentro de los SMA. En un SMA, los agentes tienen que encontrarse, anunciar sus capacidades y las tareas que pueden llevar a cabo, además de pedir o solicitar tareas a otros agentes.

### 2.2.1 Interacción entre agentes

---

En un SMA es necesario desarrollar tareas de comunicación, coordinación y negociación. Para que los agentes puedan interactuar de forma coherente, deben compartir información sobre sus objetivos y tareas. Gracias al intercambio de esta información, los agentes coordinan el desarrollo de actividades, siendo capaces de negociar en caso de que aparezcan conflictos y de planificar sus acciones para llevar a cabo un objetivo.

En las interacciones de un SMA hay que distinguir cuatro conceptos, que aunque estén muy relacionados entre sí, hacen referencia a características diferentes de un SMA. Éstos son: comunicación, coordinación, cooperación y negociación.

- ❖ **Comunicación.** Es la habilidad de los agentes para comunicarse entre sí, esto es, intercambiar información y conocimiento de forma comprensible. Permite a los agentes obtener el conocimiento necesario para decidir la secuencia de acciones que debe ejecutar en función de sus objetivos.

La comunicación posibilita la interacción entre agentes, para ello los mensajes intercambiados deben utilizar no sólo un lenguaje común, sino también deben tener la capacidad de entender e interpretar el conocimiento que se intercambia y ser capaces de intercambiarlo.

Estos lenguajes, denominados comunmente ACL (*Agent Communication Language*), están inspirados en la teoría de actos del habla [Searle, 1969] y han servido para fundamentar el repertorio de actos comunicativos básicos definidos por los estándares actuales. Los agentes pueden intercambiar mensajes entre sí mediante un lenguaje ACL normalizado. La utilización de este tipo de lenguajes es esencial si se pretende conseguir comunicar agentes pertenecientes a SMA diferentes.

La organización FIPA ha desarrollado un estándar actual llamado FIPA ACL, basado en KQML (*Knowledge Query and Manipulation Language*). Este lenguaje permite desarrollar un conjunto de 22 acciones comunicativas, que se agrupa en 4 clases, (información, realización de acciones, negociación e intermediación) y que para cada una de ellas proporciona la siguiente información:

- Una descripción detallada en lenguaje natural de la acción y sus consecuencias.
- Un modelo formal, escrito en un lenguaje semántico, que defina la semántica de la acción, sus precondiciones formales y sus efectos.
- Ejemplos de uso.

- ❖ **Coordinación.** La coordinación es una característica clave en el desarrollo de SMA. Malone [Malone, 1988] [Malone y Crowston, 1994] describe la coordinación de acciones como un conjunto de acciones suplementarias que pueden realizarse en un entorno multi-agente para alcanzar un objetivo y que un agente, con los mismos objetivos, no podría alcanzar por sí solo.

[Ferber, 1999] define 4 razones fundamentales para realizar acciones coordinadas:

1. Los agentes necesitan información y resultados que sólo pueden ser suministrados por otros agentes.
2. Los recursos son limitados. Los agentes pueden compartir recursos para poder desarrollar las acciones.
3. La coordinación permite optimizar costes, ya que elimina el desarrollo de acciones injustificadas o redundantes.
4. Los agentes tienen objetivos distintos pero interdependientes, así pues, los agentes pueden alcanzar sus objetivos beneficiándose de esta interdependencia.

- ❖ **Cooperación y negociación.** La cooperación es el mecanismo por el cual los agentes, que trabajan juntos para lograr un objetivo común, definen una estrategia para alcanzar este objetivo.

Por el contrario, la negociación modela la coordinación entre agentes auto-interesados capaces de llegar a acuerdos vinculantes. La negociación permite alcanzar decisiones de coordinación conjuntas mediante la comunicación explícita [Muller, 1996]. Estos mecanismos están inspirados en modelos tomados de las ciencias sociales y especialmente de la economía, donde prestan especial atención en la negociación estratégica y la teoría de juegos.

Los SMA se preocupan por las interacciones de los agentes que lo forman. Estos agentes forman parte de una colección y pueden coordinar su conocimiento, objetivos, habilidades y planes juntamente para tomar una acción o resolver una meta global. En todos los sistemas debe haber un proceso de racionalización para la coordinación del conjunto de agentes. Por lo general en estos sistemas, los agentes, con sus creencias, deseos e intenciones construyen el problema y el plan o secuencia de acciones para solucionarlo. La coordinación es un punto clave en el desarrollo de esta investigación. Las razones argumentadas por Ferber [Ferber, 1999] para la realización de acciones coordinadas se sustentan en el modelo presentado en el capítulo 5, haciendo que los agentes puedan compartir recursos, que se optimicen los costes del sistema y que puedan alcanzar sus objetivos y adaptarse de una manera independiente.

### 2.2.2 Adaptación en SMA

---

Como vemos, la coordinación es un pilar en el desarrollo de SMA. Es clara la tendencia actual a un desarrollo de sistemas no centralizados. Además, en la actualidad existe una notoria disposición a que la nueva generación de aplicaciones software favorezca la autonomía, robustez, flexibilidad y adaptabilidad de los sistemas a desarrollar. Para ello se hace necesario disponer de teorías, modelos, mecanismos, métodos y herramientas que permitan desarrollar sistemas con capacidad de reorganización y que puedan adaptarse de esta forma frente a futuros cambios en su entorno. En esta línea, las organizaciones dinámicas de agentes que se auto-ajustan para obtener ventajas a partir de su entorno actual son cada vez más importantes. Estas organizaciones podrían aparecer en sistemas de agentes dinámicos o emergentes, tales como las sugeridas por los dominios Grid, ambientes inteligentes u otros en los cuales los agentes se agrupan de forma dinámica para ofrecer servicios compuestos y que pueden evolucionar en el tiempo.

Los factores sociales, que permiten alcanzar acuerdos en las organizaciones de sistemas multi-agente también son cada vez más importantes para estructurar las interacciones en mundos abiertos y dinámicos. Cualquier infraestructura que dé soporte a la ejecución de aplicaciones multi-agente en estos entornos debe ser robusta, eficiente y adaptativa en el tiempo.

En definitiva, debido a las características de estos entornos abiertos, son necesarios nuevos enfoques para soportar la evolución de los sistemas, y para facilitar su crecimiento y actualización en tiempo de ejecución especialmente debido al dinamismo de estos entornos abiertos.

Las tendencias actuales en la construcción de SMA argumentan la capacidad de adaptación y aprendizaje. A la hora de tomar una decisión, los agentes no sólo tienen en cuenta la información almacenada en las bases de datos sobre situaciones previas, sino que razonan apoyándose también en las características de la situación actual.

El *Agentlink Technical Forum on Self-Organisation in MAS* [Di Marzo et al., 2004] estableció dos definiciones de sistema auto-adaptativo, una en un sentido más estricto y otra más relajada:

- i) sistemas auto-adaptativos son sistemas que cambian su organización sin ningún control central, explícito, implícito, externo ni interno;
- ii) sistemas auto-adaptativos son sistemas donde la reorganización ocurre como resultado o planificación de un control central interno.

Las diferentes propuestas para conseguir sistemas multi-agente adaptativos se pueden dividir, en función de los mecanismos utilizados, en las siguientes clases [Rodríguez et al., 2009]:

- ❖ Mecanismos basados en interacciones directas: estos trabajos utilizan los principios básicos de localización y emisión de mensajes, acompañados de interacciones y cálculos locales a los agentes, para proporcionar un estado global coherente del sistema [Zambonelli et al., 2004]. Estos trabajos se centran en el cambio de aspectos estructurales, tales como la topología de la comunicación de los agentes.
- ❖ Mecanismos basados en interacciones indirectas: estos mecanismos consiguen comportamientos complejos del sistema, a partir de las interacciones indirectas entre los agentes. Estas interacciones se producen a través de cambios producidos en el entorno [Reitbauer et al., 2004].
- ❖ Mecanismos basados en el refuerzo: este tipo de propuestas se basan en la capacidad de los agentes para modificar dinámicamente su comportamiento de acuerdo a algún refuerzo (*reinforcement*). Esto consiste en que refuerzos positivos provocan un incremento en el comportamiento de los agentes, mientras que los refuerzos negativos provocan una disminución en el mismo [Weyns et al., 2004].
- ❖ Mecanismos basados en la cooperación: este tipo de trabajos se basan en las primitivas de composición y descomposición de agentes. La descomposición conlleva la partición de un agente en dos, atendiendo a sobrecargas que puedan producirse en momentos puntuales. Por otro lado, la composición es el mecanismo inverso, por el cual dos agentes son fusionados ante una baja carga de trabajo [Caperá et al., 2003].
- ❖ Mecanismos basados en arquitecturas genéricas: estos mecanismos se basan en meta-modelos y arquitecturas de referencia de organizaciones de agentes, que son modificadas y adaptadas acorde a las necesidades de las aplicaciones en particular. Estas modificaciones dinámicas pueden ser realizadas automáticamente o por la intervención del usuario [Razavi et al., 2005].

La adaptación propuesta en el modelo social contenido en este estudio puede considerarse englobado dentro del primer tipo de mecanismo ya que utiliza interacciones y cálculos locales para reorganizar los agentes y así obtener un estado global coherente del sistema. Aún así, no debe limitarse a este tipo de funcionamiento ya que la utilización de la plataforma THOMAS [Carrascosa et al., 2009] [Giret et al., 2009], nos acerca a nuevos métodos de adaptación basados en arquitecturas en los que es posible modificar la estructura de la organización para adaptarse a los cambios.

El siguiente capítulo ahondará en estos conceptos, centrándose ya directamente en el estudio de los modelos sociales de agentes propuestos en la actualidad y cómo llevan a cabo la adaptación.

---

### 2.2.3 Sociedades de agentes

---

Hasta este momento, se ha utilizado el concepto de agente desde un punto de vista cooperativo pero siempre visto de manera individual. Es decir, se han revisado sus características como entidad software, el entorno en el que puede interaccionar y establecerse, arquitecturas, y tendencias actuales. A partir del siguiente capítulo y a lo largo de esta tesis se pondrán de manifiesto sus características sociales y se hará un análisis exhaustivo de los conceptos organizativos; pero antes, resulta necesario aclarar un concepto clave: el de "sociedad".

**Definición:** *"Una sociedad artificial está definida como un conjunto de entidades artificiales interrelacionadas e interactuantes, que se rigen bajo determinadas reglas y condiciones"*[Mauro, 2003].

El concepto de sociedad es usado ampliamente en contextos de organizaciones humanas, ecológicas o, como es este caso, de agentes. La funcionalidad principal de una sociedad es permitir a sus miembros coexistir en un entorno compartido y llevar a cabo sus respectivos objetivos cooperando o no con el resto de miembros. Las principales características que definen una sociedad son, por un lado el propósito con la que fue creada, su estructura o definición de sus partes, y por último, las reglas y normas que la rigen y que controlan el comportamiento de sus miembros. La estructura está determinada por los roles, reglas de interacción y lenguaje de comunicación entre sus miembros. De manera general podemos decir que las reglas y normas describen el comportamiento deseable de los miembros de la sociedad y establecen prohibiciones y restricciones que dan seguridad a sus miembros. La ventaja principal del desarrollo de SMA desde el punto de vista social es que permite crear sistemas con lenguajes, aplicaciones y características muy diferentes (heterogéneos).

Las organizaciones pueden verse como un conjunto de entidades reguladas por mecanismos de orden social y que persiguen objetivos comunes. Las arquitecturas que ayudan a modelar y construir SMA basados en organizaciones deberían soportar marcos de coordinación entre agentes, además de ser capaces de adaptarse dinámicamente a cambios en su estructura, objetivos o interacciones [Dignum, 2004 ].

Desde un punto de vista empresarial, el comportamiento global del sistema y los aspectos organizativos del dominio (estabilidad en el tiempo, acuerdo sobre todos los objetivos y estrategias) son muy importantes. Pero estos mismos factores, cuando hablamos desde el punto de vista de los SMA basados en

organizaciones son vitales para el sistema. Esto hace ver que los SMA pueden ser comprendidos mucho mejor si se inspiran en el comportamiento social humano [Artikis et al, 2001], [Castelfranchi, 2000], [Zambonelli et al., 2001].

Cuando los SMA son considerados desde el punto de vista social, el concepto de comportamiento deseable llega a ser de vital importancia. Es decir, el comportamiento individual de los agentes en una sociedad debe ser entendido y descrito en relación a la estructura social donde se encuentra y a los objetivos globales de la sociedad.

Hasta hace muy poco, los SMA fueron vistos principalmente desde una perspectiva individual, es decir, como agregaciones de agentes que interaccionaban unos con otros teniendo en cuenta solamente cómo afectaba el comportamiento del agente en el entorno y viceversa [Ferber y Gutknecht, 1998]. En una visión individualista de los SMA, los agentes son entidades individuales dentro de una sociedad situada en un entorno concreto, es decir, sus comportamientos dependen solamente de las reacciones al entorno y de los comportamientos de otros agentes [Dautenhahn, 2000]. No es posible imponer requisitos y objetivos a los aspectos globales del sistema, algo fundamental en los entornos empresariales. Sin embargo, las sociedades de agentes orientadas a la organización requieren una visión colectivista de la relación entre el agente y el entorno [Edmonds, 1999]. Un agente dentro de una sociedad necesita considerar no solamente su propio comportamiento sino también el comportamiento del sistema como un todo y cómo los agentes influyen unos en otros.

En [Davidsson, 2001] se propone una clasificación para las sociedades artificiales basándose en las siguientes características:

- ❖ la apertura, que describe las posibilidades de cualquier agente de unirse a la sociedad,
- ❖ la flexibilidad, que indica el grado en el que el comportamiento del agente está restringido por las normas de la sociedad,
- ❖ la estabilidad, que define la previsibilidad de las consecuencias de las acciones, y
- ❖ la fidelidad o confianza (*trustfulness*), que especifica el grado en que los de agente pueden confiar en la sociedad.

Dependiendo del propósito para el que sea creada, una sociedad necesita soportar todas estas características en diferentes grados. Por un lado, tenemos *sociedades abiertas* que no imponen restricciones a los agentes que forman la sociedad. Popper [Popper, 1982] define las sociedades abiertas, como sistemas en un estado, lejos de un equilibrio, que no muestran ninguna tendencia hacia un aumento en el desorden. Es decir, sociedades abiertas que apoyan la flexibilidad y la apertura muy bien, pero que les falta estabilidad y confianza. El ejemplo más evidente de una sociedad abierta es la WWW.



Las **sociedades abiertas** asumen que los agentes que participan son diseñados y desarrollados fuera del ámbito y el diseño de la propia sociedad y por lo tanto la sociedad no puede basarse en la incorporación de elementos organizativos y normativos en las intenciones, deseos y creencias de los agentes participantes, sino que debe representar a estos elementos explícitamente. Estas consideraciones conducen a los siguientes requisitos para las metodologías de ingeniería para la construcción de sociedades de agentes abiertas [Dignum, et al., 2001]:

- ❖ Las sociedades de agentes deben incluir formalismos para la descripción, la construcción y el control de los elementos organizativos y normativos de una sociedad (roles, normas y objetivos) en lugar de sólo los estados de los agentes [Artikis et al, 2001], [Zambonelli et al., 2001].
- ❖ La metodología debe proporcionar mecanismos para describir el entorno de la sociedad y las interacciones entre los agentes y la sociedad, y para formalizar el resultado esperado de funciones a fin de verificar la animación general de la sociedad.
- ❖ Los elementos organizativos y normativos de una sociedad deben ser especificados de forma explícita desde una sociedad abierta; no pueden depender de su inserción en las intenciones, deseos y creencias de cada agente [Dellarocas, 2000], [Ossowski, 1998].
- ❖ Son necesarios métodos y herramientas para poder comprobar si el diseño de una sociedad de agentes satisface sus necesidades de diseño y sus objetivos [Jonker et al., 2000].
- ❖ La metodología debería proporcionar directrices sobre la capacidad de comunicación y la capacidad para ajustarse al comportamiento que se espera de los agentes que participan en la sociedad.

En estas sociedades es posible entrar sin ningún control. Se pueden crear generando mínimos controles e inclusive ninguno, de esta manera se exaltan características tales como la flexibilidad y el nivel de apertura, pero por otro lado perjudican características como la estabilidad y la confiabilidad, debido a que no se tiene ningún control sobre qué entes entran a la sociedad y mucho menos sobre qué labores o acciones realizan.

Las **sociedades cerradas**, en el otro extremo, se caracterizan por tener una gran confiabilidad y estabilidad, debido a sus políticas en las que ningún agente externo puede acceder, y además las reglas que gobiernan el sistema están bien delimitadas en pro del cumplimiento de una meta específica. Los agentes en las sociedades cerradas están expresamente diseñados para cooperar hacia un objetivo común y con frecuencia se implementan de manera conjunta con la sociedad [Zambonelli et al., 2001]. Cada uno de los agentes existentes puede confiar en el otro, debido a que nunca se van a presentar extraños en la sociedad. Por otro lado, estos tipos de sociedades se caracterizan porque cada uno de los miembros persigue un mismo objetivo en común, esto no significa

que todos tengan mismas metas locales, sino que cada una de estas metas aportará en la consecución del objetivo general. Hasta hace muy poco, la gran mayoría de SMA existentes eran cerrados.

[Davidsson, 2001] introduce dos nuevos tipos de sociedades agente, semi-abiertas y semi-cerradas, que combinan la flexibilidad de las sociedades abiertas con la estabilidad de las sociedades cerradas. Este equilibrio entre la flexibilidad y la estabilidad de resultados se logra mediante mecanismos que hacen cumplir la conducta ética entre los agentes.

Las **sociedades semi-abiertas**, se determinan por tener cierto nivel de equilibrio respecto a las características anteriormente nombradas, debido a que para poder acceder a éstas es necesario, primero hacer contacto con un mecanismo de control de admisión, el cual evaluará el ente que solicita la entrada a la sociedad. Según las características que exija la sociedad, y lo que se interprete del ente se le aprobará o no el ingreso. De esta forma se permitirá el ingreso tan solo a entes que parezcan confiables y capaces ante el mecanismo decisor. Las sociedades semi-abiertas limitan ligeramente la apertura y flexibilidad de las sociedades abiertas, pero son capaces de dar una mayor estabilidad y confianza.

A diferencia de las sociedades semi-abiertas, las sociedades **semi-cerradas** y las cerradas tampoco permiten el acceso a ningún nuevo ente a la sociedad, pero por medio de un mecanismo que comunica el exterior con el interior de la sociedad se puede solicitar la creación de un nuevo agente en el interior de la sociedad, el cual funcionara en nombre del agente externo. Aunque el nuevo agente que se crea funciona como un representante del que está fuera, éste no es a la imagen y semejanza sino que es creado basado en los roles existentes dentro de la sociedad; así no tendrá ninguna característica ajena a las de los entes que pueden conformar la sociedad, y por ende el control será más efectivo [Ossowski, 1998]. Esto amplía la flexibilidad y la apertura de la sociedad, sin perder la estabilidad y confiabilidad, ya que los agentes participantes todavía están diseñados siguiendo los requerimientos de la sociedad y el propietario de la sociedad todavía controla la arquitectura general del sistema. Las sociedades semi-cerradas son tan abiertas como las sociedades semi-abiertas, pero menos flexibles. Por ejemplo, este es el enfoque adoptado en la plataforma ISLANDER donde los agentes externos proporcionan un API como interfaz para la institución, que regula y controla toda la interacción [Esteva et al., 2002].

Las organizaciones de agentes de tipo abierto son las que más auge están teniendo en las investigaciones actuales. De hecho, las organizaciones de agentes u organizaciones virtuales de agentes (VO) como también son llamadas, pueden considerarse sistemas abiertos formados por la agrupación y colaboración de entidades heterogéneas y donde existe una separación clara entre estructura y funcionalidad definiendo cómo se comportan las entidades

[Foster et al., 2001][Boella et al., 2005]. Como se verá en la sección de este estudio centrada en los enfoques sociales (sección 3), muchas de las investigaciones recientes no sólo se centran en el empleo de estructuras organizacionales durante el proceso de diseño, sino también en la regulación y adaptación en SMA abiertos [Esteva et al., 2002] [Noriega y Sierra, 2002][Hubner, 2004].

---

#### 2.2.4 Tendencias

---

La evolución de los agentes es constante, tratando de adaptarse en cada momento a las tecnologías emergentes de la información. La aparición de la Inteligencia Ambiental [Richter, K. & Hellenschmidt, M., 2004] o de entornos Inteligentes [Vázquez and López de Ipiña, 2005] refleja la importancia que adquiere esta tecnología en la vida diaria de las personas. Por ejemplo:

- ❖ Web Semántica. Definida como una extensión de la web actual en la que se proporciona información con un significado bien definido, y se mejora la forma en la que las máquinas y las personas trabajan en cooperación [Berners-Lee et al., 2001]. La potencia real de la Web Semántica se aprecia cuando existen agentes capaces de manejar contenido semántico que son usados para recoger y procesar información Web e intercambiar los resultados con otros agentes. Así pues, el desarrollo de la Web semántica está íntimamente ligado a la evolución de la tecnología de agentes [Berners-Lee et al., 2001].
- ❖ Servicios Web. Los Servicios Web permiten publicar componentes y servicios en un directorio o registro de servicios en el que otras aplicaciones Web pueden buscar e implementar nuevos servicios a través de una llamada [Samper, 2005]. Por lo tanto, los servicios Web proporcionan una infraestructura ideal para el desarrollo de las interacciones de los sistemas multi-agente. La aproximación orientada a agentes de los servicios web está ganando crédito, ya que cada vez más los entornos de servicios web son vistos como sistemas basados en agentes [Booth et al, 2004].
- ❖ Computación *peer-to-peer*. Este tipo de tecnología cubre un amplio rango de infraestructuras, tecnologías y aplicaciones que tienen una característica en común: están diseñados para la creación de aplicaciones en red, en la que todos los elementos del sistema (nodos) son iguales. Así pues, en estos sistemas no es necesario la existencia de un servidor centralizado, por lo que la tecnología de agentes se presenta como un entorno tolerantes a fallos, escalables y fácilmente implementables y mantenibles [Milojicic et al., 2002].
- ❖ *Grid computing*. El *grid* es una infraestructura de computación de alto rendimiento que permite desarrollar tareas con elevados costes

computacionales [Kaufmann, 2004]. Así pues, el grid alude a la infraestructura que facilita la colaboración de computadores de altas prestaciones, de redes, de bases de datos, de agentes, y en general de instrumentos científicos, que pertenecen y son gestionados por múltiples organizaciones [Luck et al., 2005].

- ❖ Inteligencia ambiental. El término “*Ambient Intelligence (AmI)*” surge en 1999 como una propuesta realizada por el *Information Society Technology Programme Advisory Group (ISTAG)* de la Comunidad Europea [Richter, K. & Hellenschmidt, M., 2004]. El objetivo de esta tecnología se centra en la integración de servicios y aplicaciones, para lo que se apoya en tareas como la computación ubicua y las interfaces de usuario inteligentes. El enfoque de estos sistemas se centra en extensos entornos compuestos, potencialmente, por una gran cantidad de elementos interactuando entre sí para facilitar el logro de metas y actividades. Las características necesarias para poder integrar estos servicios, (autonomía, distribución, adaptabilidad, y capacidad de reacción) son también factores clave en los sistemas multi-agente.
- ❖ Sistemas autónomos. Estos sistemas se centran en aquellos entornos computacionales que son capaces de gestionarse a sí mismos. Esta emergente tecnología se basa en aspectos como autoorganización, autoconfiguración, autogestión, autodiagnóstico y autocorrección [Luck et al., 2005], características muy investigadas de los agentes y SMA.

En estos tipos de entornos deben vivir los agentes, situándose en distintos dispositivos de diferentes características y conviviendo con otros agentes, con personas y con el resto de elementos inteligentes del entorno. La noción de entorno ha sido muy importante en la definición de sistema multi-agente. Es importante obtener abstracciones que permitan estudiar y modelar el entorno en el que se desarrolla un SMA.

Hace no muchos años, los SMA estaban diseñados por equipos individuales o entornos de trabajo corporativos. En estos sistemas, que se desarrollaban para un dominio particular, participaban un conjunto de agentes que compartían un conjunto de objetivos. Estos sistemas eran escalables únicamente en entornos controlados o simulados. Los lenguajes de comunicación y los protocolos de interacción estaban diseñados a medida. Es decir, estaban diseñados por los desarrolladores antes de que el agente fuese ejecutado e interaccionara con otros agentes. Las plataformas de desarrollo, así como las técnicas de diseño y modelado, se realizaban a medida, inspiradas por el paradigma orientado a agentes, más que por la utilización de metodologías estandarizadas [Luck et al., 2005].

Actualmente, los SMA ya permiten la participación de agentes heterogéneos, diseñados por diferentes equipos y organizaciones. Cualquier agente debe ser capaz de participar en estos sistemas, desarrollando comportamientos públicos y estandarizados. Sin embargo, estos sistemas suelen seguir siendo desarrollados para dominios de aplicaciones particulares. Los sistemas desarrollados son implementados con metodologías específicas y estandarizadas que incluyen plantillas y patrones para diferentes tipos de agentes y organizaciones. El desarrollo de lenguajes de programación y herramientas específicas se extiende, lo que permite la aplicación de técnicas de especificación formal. Los elementos semánticos tienen una gran importancia, por ejemplo para realizar acciones coordinadas entre agentes heterogéneos.

En un futuro a muy corto plazo (la tendencia en las investigaciones actuales va en esta línea), se desarrollarán SMA en dominios abiertos que involucrarán la participación de agentes heterogéneos diseñados por diferentes equipos u organizaciones. Los agentes, que participarán en estos sistemas, serán capaces de aprender los comportamientos adecuados para participar en cualquier interacción que se pueda presentar durante su ejecución. Además, la identificación de protocolos de interacción y comunicación, serán seleccionados automáticamente.

Lo que hace no mucho se veía como un futuro a medio y largo plazo [Luck et al., 2005] es ahora casi una realidad: agentes que pueden formar o pertenecer a coaliciones que se crean dinámicamente y cuyas propiedades (positivas y negativas) no están definidas a priori, sino que emergen a lo largo del proceso de ejecución de los agentes.

### 3 MODELOS SOCIALES PARA SISTEMAS ADAPTATIVOS Y SUS ENFOQUES

---

En el anterior capítulo se han descrito los conceptos básicos necesarios para comprender el paradigma de agentes, sus interacciones, sus características planificadoras y se ha adelantado el concepto de "sociedad de agentes". Ahora, la revisión tecnológica se centrará en las características sociales de los SMA. Entre ellas, conceptos esenciales como organización virtual, modelo social, estructura social, normas, roles, etc. que ayudan a comprender la finalidad de este trabajo. La coordinación adaptativa se lleva a cabo dentro de una sociedad y por tanto es necesario hacer una revisión de las posibilidades, estudios e investigaciones actuales dirigidas en esta dirección. Para ello, se ha realizado un estudio profundo de los modelos sociales de agentes existentes y de cómo cada uno de ellos lleva a cabo la coordinación y las posibles adaptaciones.

#### 3.1 ORGANIZACIONES DE AGENTES

---

¿Existe una analogía real entre las comunidades humanas y las organizaciones formadas por agentes? Mientras se buscan respuestas a esta pregunta, el desarrollo de este tipo de sistemas va avanzando. Existe un amplio abanico de investigaciones enfocadas a la construcción de organizaciones virtuales, a la simulación social basada en agentes, al estudio del comportamiento, etc. [Sansores y Pavón, 2005][Iglesias, 2010]. Está claro que nos dirigimos hacia un modelo de computacional gremial.

Debido a esta analogía, podemos definir "organización" desde 2 puntos de vista: desde el punto de vista humano y desde el de la tecnología de agentes:

Organizaciones humanas:

- ❖ **Definición:** *"Asociación de personas regulada por un conjunto de normas en función de determinados fines"* (Real Academia de la lengua española).
- ❖ **Definición** *"Formación o entidad social con un número de miembros que puede ser precisado y una diferenciación interna de las funciones que son desempeñadas por dichos miembros"* [Peiro,1990].

Organizaciones de agentes:

- ❖ **Definición:** *"Una organización proporciona un marco de trabajo para la actividad e interacción de los agentes a través de la definición de roles, expectativas de comportamiento y relaciones de autoridad, como el control."* [Gasser & Ishida 1991].

- ❖ **Definición:** *“La organización es una colección de roles, que mantienen ciertas relaciones entre sí, y que toman parte en patrones de interacción con otros roles de forma institucionalizada y sistemática” [Zambonelli et al. 2003].*

La teoría estructuralista [Taylor, 1991], primera en defender la "organización científica del trabajo" modelaba las organizaciones como sistemas cerrados. Los sistemas son cerrados cuando están aislados de variables externas y cuando son determinísticos. Un sistema determinístico es aquel en que un cambio específico en una de sus variables produce un resultado particular con certeza. Así, el sistema requiere que todas sus variables sean conocidas y controlables o previsibles. La eficiencia organizacional en estos tipos de sistemas, siempre prevalecerá si las variables organizacionales son controladas dentro de ciertos límites conocidos.

Sin embargo, las organizaciones virtuales poseen todas las características de los sistemas abiertos. Algunas características básicas en las que las organizaciones son vistas como sistemas abiertos son [Spencer, 1896]:

- ❖ *En el crecimiento.*
- ❖ *En el hecho de volverse más complejas a medida que crecen.*
- ❖ *En el hecho de que haciéndose más complejas, sus partes exigen una creciente interdependencia.*
- ❖ *Porque su vida tiene mayor extensión comparada con la vida de sus unidades componentes.*
- ❖ *Porque en ambos casos existe creciente integración acompañada por creciente heterogeneidad.*

Cambio, interdependencia y heterogeneidad son conceptos de los sistemas abiertos que se manifiestan también en las sociedades.

Schein [Schein 1985] propone una relación de aspectos que deberían considerarse en la creación de una organización y que definen en su conjunto el concepto de "modelo social" desde una perspectiva general:

- ❖ La organización debe ser considerada como un sistema abierto.
- ❖ La organización debe ser concebida como un sistema con objetivos o funciones múltiples.
- ❖ La organización debe ser visualizada como una constitución de muchos subsistemas que están en interacción dinámica unos con otros.
- ❖ Al ser los subsistemas mutuamente dependientes, un cambio en uno de ellos, afectará a los demás.
- ❖ La organización existe en un ambiente dinámico que comprende otros sistemas.
- ❖ Los múltiples eslabones entre la organización y su medio ambiente hacen difícil definir las fronteras de cualquier organización.

Los modelos sociales defienden el diseño de SMA inspirado en conceptos y teorías sociales como normas, convenciones (o costumbres) sociales u organizaciones. Por este motivo, la estructura organizacional es adecuada para diseñar mecanismos de coordinación en SMA [Gasser et al., 1987][Pattison et al., 1987].

Centrándonos en el paradigma computacional, los sistemas pueden ser vistos, de una forma natural, en términos de entidades (comúnmente agentes) que proveen y consumen recursos [Luck et al., 2008], que probablemente han sido diseñadas por equipos de desarrollo diferentes y que pueden entrar o salir de una organización en diferentes momentos y debido a diferentes razones. Además, pueden formar coaliciones u organizaciones entre ellas y tener unos mismos objetivos. Las tendencias actuales conducen claramente al paradigma de las Organizaciones Virtuales (OV) [Ferber et al., 2004].

**Definición:** *"Una Organización Virtual es un conjunto de individuos e instituciones que necesitan coordinar sus recursos y servicios dentro de unos límites institucionales"* [Foster et al., 2001] [Boella et al., 2005].

Por lo tanto, una OV es un sistema abierto formado por la agrupación y la colaboración de entidades heterogéneas y donde hay una separación entre la forma y la función que define su comportamiento.

La tecnología multi-agente, la cual permite la formación dinámica de organizaciones de agentes, es particularmente adecuada para el desarrollo de este tipo de sistemas. El modelado de organizaciones basadas en SMA abiertos no sólo hacen posible describir la composición estructural del sistema (por ejemplo roles, agentes, grupos, tareas, planes o servicios), sino también las normas y reglas para el control del comportamiento de los agentes, la entrada/salida dinámica de los componentes y la formación, también dinámica, de grupos de agentes. Las investigaciones en organizaciones de agentes varían desde los conceptos básicos como grupos, comunidades, roles, etc. [Jennings y Wooldridge, 2002] [Zambonelli y Parunak, 2002] [Ferber et al., 2004], pasando por el modelado de organizaciones [Horling y Lesser, 2004] [Ferber et al., 2004] [Dignum et al., 2005], la Teoría de la Organización Humana [Argente et al. 2006], topologías estructurales [Horling y Lesser, 2004] [Argente et al. 2006], hasta investigaciones normativas, incluyendo la representación de normas [López et al., 2006], lógicas deontológicas [Dignum y Dignum, 2007] o enfoques institucionales [Esteva, 2003] [Esteva et al., 2001]. El desarrollo de SMA abiertos es actualmente un campo de investigación reciente en el paradigma de agentes y su desarrollo permitirá aplicar esta tecnología en nuevos y complejos dominios de aplicación [Bajo et al. 2008] [Hernández et al. 2006] [Jiao y Mei, 2004]. En este sentido es necesario investigar en nuevos métodos para modelar organizaciones virtuales basadas en SMA abiertos e innovar en técnicas que proporcionen capacidades a las organizaciones virtuales.



### 3.1.1 *Conceptos organizativos*

---

Siguiendo la perspectiva organizacional, un sistema es descrito por una estructura social y un conjunto de normas que regulan la interacción entre agentes. Tal descripción identifica los componentes funcionales del sistema (agentes), sus responsabilidades (tareas que tiene que conseguir), y recursos (conocimiento, software, hardware, herramientas, etc.), y la relación entre ellos (comunicación, asignación, etc.). Todos estos conceptos organizacionales son esenciales para comprender las sociedades de agentes. A continuación de describen algunos de ellos.

#### **Entidad social**

Las organizaciones están formadas por componentes o entidades sociales que pueden estar compuestos a su vez por un número específico de miembros o agentes. Estas entidades, según [Pattison et al., 1987]:

- ❖ tienen unas responsabilidades, es decir, un conjunto de sub-tareas que llevar a cabo, incluidas dentro de los objetivos de la organización,
- ❖ tienen y consumen recursos. Los componentes poseen ciertos recursos con los cuales llevan a cabo sus tareas. Los recursos requeridos por un componente dependerán del rol que esté jugando en ese momento en la organización,
- ❖ están estructurados siguiendo determinados patrones de comunicación,
- ❖ tratan de alcanzar los objetivos globales de la organización y
- ❖ están reguladas por normas y restricciones.

#### **Estructura**

Las entidades en una organización no son independientes unas de otras. Interactúan entre sí. Comandos, información, etc. son unidades pasadas entre ellas. Estas interacciones son expresadas como relaciones entre los componentes. En general, estas relaciones no vienen dadas de manera individual dentro de una organización, sino que se requiere una conjunción de relaciones entre grupos de entidades.

En esta conjunción entran en juego distintos aspectos a considerar: roles, topología, relaciones de autoridad. Todos ellos determinarán la estructura de una organización.

La estructura puede definirse como la distribución, orden e interrelación de las diferentes partes que componen la organización. En esa estructura los agentes se ordenarán y comunicarán dependiendo de la topología que defina esta estructura. Existen diferentes topologías de organización, entre ellas:

- ❖ **Jerarquías:** los agentes se ordenan en una estructura de tipo árbol en la que los niveles inferiores tienen la funcionalidad básica y los niveles superiores la toma de decisiones y el control.
- ❖ **Holarquías:** son estructuras anidadas y jerárquicas de *holones*. Un holón es una parte de entidad mayor resultado de la agrupación de entidades subordinadas. Este tipo de topología suele aplicarse en dominios donde los objetivos se descompongan de forma recursiva en subtareas.
- ❖ **Coaliciones:** son agrupaciones temporales de agentes para la consecución de un objetivo concreto, que suele obtener ciertos beneficios y reducir costes. Las coaliciones se disuelven al alcanzar el objetivo, al no existir ya la necesidad de agrupación o cuando una cantidad crítica de agentes abandona la agrupación. Internamente suele representarse como una estructura plana o con líder (representante grupo) y externamente como una entidad única y atómica.
- ❖ **Grupos:** son agrupaciones de agentes cooperativos que trabajan juntos en la consecución de un objetivo común. De esta manera, maximizan la utilidad del equipo. La representación de objetivos, creencias y planes se lleva a nivel de equipo. Los grupos suelen aplicarse cuando resolver problemas planteados puede conseguirse de una mejor manera de forma conjunta. Los grupos implican una mayor redundancia y flexibilidad para entornos inciertos, aunque también un aumento en las comunicaciones (por la coordinación).
- ❖ **Congregaciones:** son agrupaciones de agentes con características similares o complementarias. En este caso, no conllevan consecución de un objetivo específico pero sí facilitan la búsqueda de colaboradores adecuados para lograr alcanzar ese objetivo. Por este motivo, este tipo de topología suele ser pensada para objetivos a largo plazo.
- ❖ **Federaciones:** son agrupaciones de agentes con un representante. Los miembros del resto de la organización interactúan solo con el representante, ceden parte de su autonomía. Este agente "representante" actúa también de intermediario entre el grupo y el mundo exterior, haciendo funciones de: (i) *Broker*: distribuye tareas entre miembros del grupo; (ii) *Mediador*: facilita interacciones entre diferentes agentes (establece contactos); (iii) *Monitor*: controla estados de los agentes, informa sobre eventos; (iv) *Embassy*: controla la comunicación de agentes externos con los de la federación (traductor de ontologías).
- ❖ **Organizaciones matriciales:** en este tipo de topología de organización, un agente puede ser controlado por más de un agente supervisor. Por este motivo, es necesario la utilización de mecanismos de evaluación de compromisos y resolución de conflictos locales. Es como una estructura de tipo grid en la que los agentes *Managers* se sitúan alrededor de agentes.

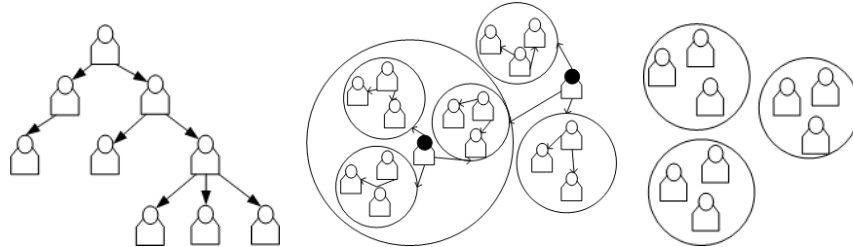


Figura 4. Topología de organización. De izquierda a derecha: jerarquía, holarquía, coaliciones.

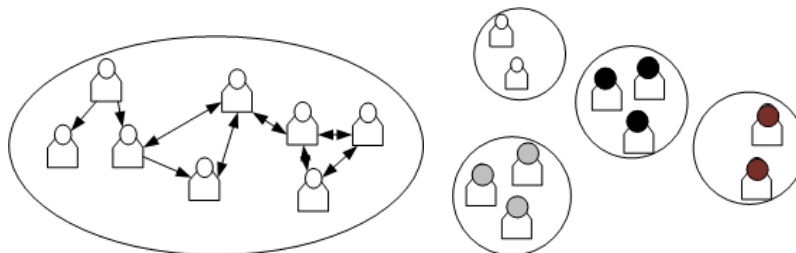


Figura 5. Topología de organización. De izquierda a derecha: grupos, congregaciones.

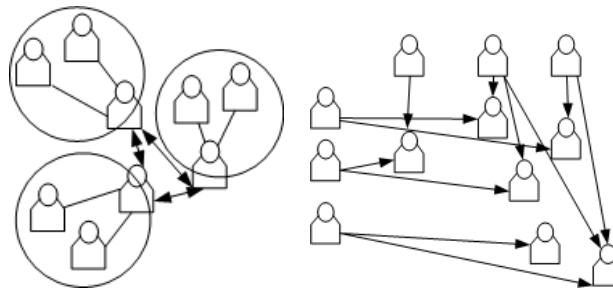


Figura 6. Topología de organización. De izquierda a derecha: federaciones, organizaciones matriciales.

### **Funcionalidad**

La funcionalidad de una organización está determinada por su misión, es decir, por los objetivos globales que describen la finalidad de su propia existencia. La misión define la estrategia, los requerimientos funcionales (qué hace la organización) y de interacción (cómo lo hace).

Los objetivos podemos clasificarlos en :

- ❖ Funcionales: de cada grupo o unidad organizativa.
- ❖ Operativos: de los agentes, sus planes (tareas que llevarán a cabo).

### **Normatividad**

Las normas sociales definen las consecuencias de las acciones de los agentes:

- ❖ Restricciones sobre la organización.
- ❖ Obligaciones, sanciones a aplicar.
- ❖ Control de accesos externos.
- ❖ Descomposición:
  - Acciones que provocan la activación de la norma.
  - Conjunto de obligaciones que adquiere el agente.
  - Acciones que se deben llevar a cabo para eliminar la obligación.

### **Entorno**

El entorno define qué existe alrededor del sistema: recursos, aplicaciones, objetos, supuestos, restricciones, *stakeholders* (proveedores, clientes, beneficiarios). Definiendo el entorno se establece la relación de los roles respecto a los elementos del entorno: modo de acceso (lectura, interacción, extracción de información), permisos de acceso, etc.

### **Dinamicidad**

La dinámica organizativa está relacionada con la entrada/salida de agentes, con la adopción de roles por parte de los mismos, la creación de grupos y con el control del comportamiento.

En la definición de la dinámica de una organización habrá que especificar:

- ❖ Respecto a la entrada de agentes: cuándo permitir a los agentes entrar en la organización; cuál será su posición en la organización; los procesos de expulsión de agentes con comportamientos anómalos. Por ejemplo: en el modelo definido por [Esteva et al., 2002] existe un agente *Institution Manager* que autoriza a agentes externos a entrar en la institución.
- ❖ Respecto a la adopción de roles: cómo los agentes adoptan un rol concreto; la asociación de agentes a uno o más roles. Por ejemplo, en [Esteva et al., 2002], se describen las transiciones entre escenas en función de roles; cambios de roles.
- ❖ Respecto a la creación dinámica de grupos: la definición de federaciones, coaliciones, congregaciones, etc.
- ❖ Por último, respecto al control del comportamiento: cómo controlar la conformidad del comportamiento de los agentes a las normas de la sociedad. Por ejemplo, en [Esteva et al., 2002] existe una capa social que garantiza que las interacciones se producen de acuerdo a las normas.

### **Adaptación social**

La adaptación para una sociedad es su capacidad para involucrarse con el entorno y hacer parte de éste una simbiosis que permita a ambos disponer uno del otro. Y no sólo puede considerarse como una capacidad, sino también como una necesidad de involucrarse al entorno para aprovechar al máximo las necesidades de aprendizaje de cada individuo y que llevará al sistema a adquirir un aprendizaje significativo.

### **Aprendizaje social**

El aprendizaje social se percibe como un proceso en el cual, gracias a un ambiente o entorno en común (proveído por una sociedad artificial), diferentes entes pueden interactuar y evaluar sus experiencias e información, teniendo en cuenta que no es un proceso crítico de un ente hacia otro en donde se determina si este tiene información correcta o no [Duong y Grefenstette, 2005], sino que cada miembro de la sociedad artificial observa al otro como una simple fuente de datos, en donde según la función de utilidad o el objetivo propuesto para el ente que está aprendiendo se define la pertinencia de los datos encontrados y de ésta manera decide que aprender y que no [Conte y Paolucci, 2001]. Un ente que aprende de otro ente, está ahorrando mucho esfuerzo en experimentación y tiempo, debido a que la información que captó es obtenida mediante el esfuerzo de otros. Finalmente, se espera lograr una unificación de conocimientos, en donde la mayoría de entes converjan al aprendizaje más cercano al ideal [Conte y Paolucci, 2001]. Los dos tipos de aprendizaje en sociedad más relevantes son:

- ❖ **Facilitación social:** en donde el proceso de aprendizaje actualiza la base de conocimiento del agente que obtiene la nueva información por medio de “observación” de lo que sucede a otro agente, mientras éste interactúa con el medio. Así, el agente artificial aprenderá según la manera en que el entorno interactúe sobre el agente observado tal como lo muestra la Figura 7. Éste tipo de aprendizaje es muy común en sociedades artificiales de organismos en donde al observar las repercusiones del entorno en respuesta a las acciones de un agente; los demás tendrán un punto de referencia para saber qué utilidad representan determinados comportamientos [Mataric, 1997]. Un ejemplo muy común de esto es en donde en el entorno se pone comida envenenada y al ser consumida los organismos artificiales mueren inmediatamente, lo cual es un factor negativo respecto a su función de utilidad o el objetivo que persiguen; al observar esto, los demás organismos evitarán consumir el tipo de alimento “etiquetado” como envenenado.
- ❖ **Aprendizaje imitativo:** en donde el agente que aprende también tiene en cuenta la experiencia que recibe del medio después de interactuar con él; es decir, después de que el agente percibe las acciones de otro

agente que interactuó con el medio, éste no tomará ésta información como una base 100% cierta, sino que primero la probará imitando la situación presentada en la experimentación, entonces finalmente lo que se tendrá será, un dato generado por un agente cualquiera contra un dato generado por él mismo [Conte y Paolucci, 2001]. Después de haber realizado el mismo experimento, de ésta manera se verificará y se tendrá una mayor seguridad acerca de la validez o correctitud de la información captada inicialmente. Es importante aclarar que el hecho de imitar otro ente va ligado directamente al nivel de “confianza” que se tenga sobre éste, tal como muestra la Figura 7, el nivel de confianza depende de aspectos inherentes a los objetivos buscados por el ente, los cuales hasta determinado punto, son muy relativos; por ejemplo, el hecho de que los entes tengan diferentes funciones objetivo no lo hace atractivo para imitar; pero si tal vez éstas metas siendo diferentes tienen ciertos puntos en común, podrían ser imitadas. Ahora el hecho se basaría en saber cómo determinar cuánto de similares podrían llegar a ser diferentes objetivos. Éste proceso imitativo es uno de los más usados en simulaciones de entornos padre e hijo, en donde los hijos de un padre aprenden de él mediante imitación; padre e hijo tienen mismas funciones objetivo, por lo tanto el nivel de confianza llega a ser total y de ésta manera el proceso imitativo no tendría en cuenta el decidir qué tan conveniente es imitar o no a otro ente.

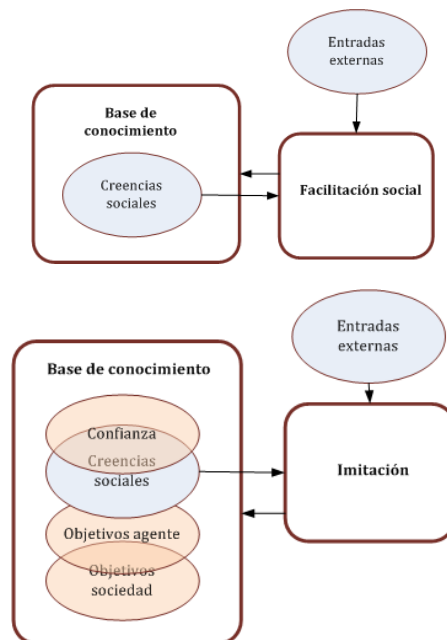


Figura 7. Aprendizaje por facilitación social. Aprendizaje por imitación

El aprendizaje como vemos, está intrínsecamente relacionado con la adaptación. Tanto la facilitación social como la imitación son técnicas que nos permiten aprender de nuevas situaciones para poder actuar de cierta manera en el futuro. En definitiva, eso puede considerarse un proceso de adaptación a nuevas situaciones. Como veremos en la sección 3.3, existen diferentes enfoques de adaptación en sociedades, y entre ellos la imitación es uno de los más utilizados [Conte y Paolucci, 2001].

---

### 3.1.2 Arquitecturas

---

Las organizaciones virtuales pueden considerarse sistemas abiertos formados por la agrupación y colaboración de entidades heterogéneas y donde existe una separación clara entre estructura y funcionalidad [Foster et al., 2001][Boella et al., 2005]. Así podemos encontrar trabajos enfocados al desarrollo de nuevas metodologías y procedimientos de diseño centrados en los aspectos de organización de los SMA como Gaia [Zambonelli et al., 2003], AGR [Ferber et al., 2004], MOISE [Hubner, 2004], OperA [Dignum, 2004] (based on ISLANDER [Esteve et al., 2001] framework), Tropos [Bresciani et al., 2004], PASSI [Cossentino, 2005], SODA [Molesini et al., 2006] MenSA [Ali et al., 2008] , O-MASE [DeLoach, 2009], INGENIAS [Pavón et al., 2005] and VOM [Criado et al., 2009]. Muchos de los estudios recientes no solo se centran en el empleo de estructuras organizacionales durante el proceso de diseño, sino también en la regulación y adaptación en SMA abiertos.

Aunque algunas de las plataformas hacen frente a los conceptos de organización mediante patrones de diseño y técnicas similares, la mayoría no pueden ser aplicadas directamente en el desarrollo de SMA abiertos donde las estructuras organizacionales pueden emerger y cambiar dinámicamente en tiempo de ejecución. Las abstracciones y herramientas actualmente disponibles todavía no son lo suficientemente potentes para desarrollar sistemas abiertos que traten con problemas dinámicos reales. La mayoría de los trabajos referenciados son, en cierto modo, incompletos, ya que no incluyen todas las etapas y las necesidades de todo el desarrollo de sistemas de este tipo. Ninguno de ellos tiene en cuenta los problemas asociados con el desarrollo de SMA abiertos en las últimas fases, en las que el ingeniero del software debe traducir las propiedades de la organización, especificadas en el análisis, y los modelos de diseño a código ejecutable en las plataformas de agentes.

El principal problema para la implementación de una Organización Virtual es la falta de plataformas que dan soporte a este tipo de sistemas. La función primordial de una plataforma de agentes es ofrecer un entorno de ejecución para los agentes. En los últimos años, han aparecido enfoques de investigación intentando ofrecer un entorno para este tipo de sistemas. Ejemplos de plataformas de agentes son JADE<sup>7</sup> , FIPA-OS<sup>8</sup> , RETSINA [Giampapa and Sycara,

---

<sup>7</sup> <http://jade.tilab.com/> [Último acceso 20/01/2010]

2002] Grasshopper [Baumer et al., 2000], Jack [Howden et al, 2001], ZEUS [Hyancinth et al., 1999], MadKit [Gutknecht y Ferber , 1997], EIDE [Esteva et al., 2002], RICA-J [Serrano y Ossowski, 2004], S-Moise+ [Hubner et al., 2006], Jack Teams (una extensión de JACK) [Agent-Oriented-Software, 2004], SIMBA [Carrascosa et al., 2003] y SPADE [Escriva et al., 2006].

Durante el desarrollo de este trabajo, se analizaron las propiedades de las diferentes propuestas. Una comparación de las mismas puede verse en el capítulo 7. Adicionalmente, un estudio comparativo puede encontrarse en [Argente et al., 2004]. Aunque la mayoría de las plataformas nombradas tienen en cuenta aspectos organizacionales, la mayoría lo hace en fases de diseño y análisis y no pueden utilizarse directamente para el desarrollo de SMA abiertos donde las estructuras organizacionales pueden emerger y cambiar dinámicamente en tiempo de ejecución.

Estas características, junto con un entorno específicamente diseñado para la ejecución de organizaciones virtuales son presentadas en la arquitectura THOMAS *'MeTHods, Techniques and Tools for Open Multi-Agent Systems'* [Carrascosa et al., 2009] [Giret et al., 2009]. THOMAS, es la arquitectura utilizada en este trabajo y sobre la que se desarrolla el modelo adaptativo de organización propuesto. La siguiente subsección muestra las características principales de la misma.

### 3.1.2.1 THOMAS

THOMAS<sup>9</sup> surge por la necesidad de dar soporte a la necesidad de desarrollar arquitecturas con las características expuestas en la sección anterior para el desarrollo de sistemas multi-agente abiertos desde el punto de vista organizativo [GTI-IA, 2009]. La arquitectura presenta la infraestructura necesaria para emplear los conceptos de la tecnología de agentes en el proceso de desarrollo, aplicando las técnicas de descomposición, abstracción y organización y teniendo en cuenta todos estos requerimientos. Es sobre THOMAS donde se propone el modelo de este estudio para permitir la evaluación del comportamiento de una organización virtual de tal manera que los agentes puedan adaptarse y reorganizarse dinámicamente.

Podríamos definir varias áreas de investigación en las que se fundamenta la construcción de la arquitectura THOMAS; entre ellas destacan la computación orientada a servicios, la estructura organizacional de las comunidades y las plataformas de agentes.

La arquitectura está formada básicamente por un conjunto de servicios estructurados modularmente. THOMAS toma como base la arquitectura FIPA<sup>10</sup>,

---

<sup>8</sup> <http://fipa-os.sourceforge.net/> [Último acceso 20/10/2010]

<sup>9</sup> <http://thomas-tin.usal.es/> [Último acceso 20/01/2010]

<sup>10</sup> <http://www.fipa.org> [Último acceso 20/10/2010]



expandiendo sus capacidades respecto al diseño de organizaciones e impulsando la capacidad de los servicios. En THOMAS, existe un módulo con el único objetivo de manejar las organizaciones que hayan sido introducidas en la arquitectura y es introducida una redefinición del *FIPA Directory Facilitator*, capaz de hacer frente a los servicios de un modo más elaborado.

Los agentes acceden a la infraestructura ofertada por THOMAS por medio de una serie de servicios incluidos en lo que se denomina OMS (*Organization Manager Service*). Los componentes principales de THOMAS podemos verlos en la Figura 8 (adaptada de [GTI-IA, 2009]):

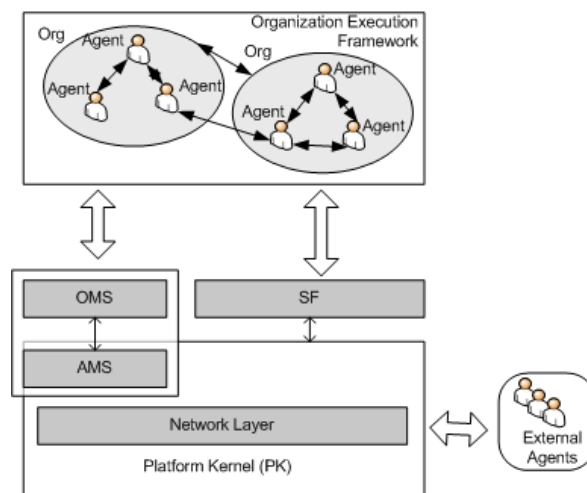


Figura 8. Arquitectura THOMAS

Como se observa, existen tres componentes principales en THOMAS:

- *Service Facilitator (SF)*: este componente ofrece servicios simples y complejos a los agentes y organizaciones activas. Básicamente su funcionalidad se resume en ofrecer un directorio de páginas amarillas y verdes sobre los servicios disponibles.
- *Organization Manager Service (OMS)*: principalmente se encarga de la gestión de organizaciones y de las entidades incluidas en ellas. Permite por tanto la creación y gestión del ciclo de vida de una organización.
- *Platform Kernel (PK)*: permite mantener los servicios básicos de gestión de una plataforma de agentes.

### **Service Facilitator (SF)**

Este componente ofrece el soporte necesario para que los agentes y las organizaciones ofrezcan y descubran servicios. Principalmente, el SF provee un

lugar donde las entidades autónomas pueden registrar la descripción de servicios como entradas en un directorio. Es decir, su función es actuar como un gestor de páginas amarillas y de esta manera se pueden hacer búsquedas para determinar qué entidades proveen un determinado servicio. El SF proporciona un lugar donde las entidades autónomas pueden registrar la descripción de servicios como entradas en un directorio. Gracias a esto, podrá localizar servicios de acuerdo a sus perfiles o a los objetivos que se desean satisfacer mediante su invocación, incorporando mecanismos de composición de servicios.

El SF será el que controle el acceso a la plataforma THOMAS, utilizando técnicas de seguridad y gestión de permisos.

Un servicio es caracterizado normalmente como la interacción entre dos entidades, y es modelado como comunicaciones entre procesos independientes. En general, un servicio ofrece unas capacidades, cada una de ellas encaminadas a cumplir un objetivo, que necesita que se verifique ciertas condiciones antes de su ejecución y que intercambian uno o más mensajes de entrada y salida y que tras su ejecución con éxito tiene una serie de efectos en el entorno. Además, existen parámetros adicionales, independientes de su funcionalidad, como la calidad del servicio, plazos o protocolos de seguridad, entre otros (parámetros no funcionales). En último lugar, los resultados obtenidos por los servicios pueden ser mejorados usando mecanismos de composición automática de servicios [Brogi et al., 2003] (como por ejemplo, *matchmaking* parcial). Para esto, el SF tiene una descripción de los procesos internos que tienen lugar cuando se ejecuta el servicio. En THOMAS, al disponer de los protocolos de FIPA existe un mecanismo bien establecido para estandarizar las interacciones. Así, cada servicio tiene asociado un protocolo. Hay que considerar que estamos hablando de servicios semánticos, por lo que será importante la ontología utilizada en el servicio que permita, al leer la descripción de un servicio, que cualquier entidad obtenga toda la información necesaria para saber cómo interactuar con él o cómo construir una aplicación que sea capaz de utilizarlo

Un servicio es definido como una tupla del tipo  $(sID, goal, prof, proc, ground, ont)$ , donde:

- ❖ *sID* es el identificador único del servicio.
- ❖ *goal* es el objetivo, el propósito que el servicio pretende conseguir y proporciona el primer nivel de abstracción para la composición del servicio.
- ❖ *prof* es el perfil del servicio y lo describe en términos de su *IOPEs* (*Inputs, Outputs, Preconditions and Effects*) (Entradas, Salidas, Precondiciones y Efectos) y sus atributos no funcionales. Este tipo de representación incluye una descripción de lo que cumple el servicio, las limitaciones a su aplicabilidad y la calidad del servicio, y los requisitos que deben cumplir los clientes para poder utilizar el

servicio. Este tipo de representación incluye una descripción de lo que cumple el servicio, las limitaciones a su aplicabilidad y la calidad del servicio, además de los requisitos que deben cumplir los clientes para poder utilizar el servicio.

- ❖ *proc* describe cómo debe usar el servicio un cliente. Se especifica el contenido semántico para el uso del servicio, las situaciones en las que se obtiene, y, cuando sea necesario, el proceso paso a paso para obtener estos resultados. En otras palabras, se especifica cómo llamar a un servicio y lo que ocurre cuando el servicio se ejecuta .
- ❖ *ground* especifica en detalle cómo puede un agente acceder al servicio. El *grounding* especifica un protocolo de comunicación, los formatos de mensaje, el puerto de contacto y otros detalles del servicio. Se especifica mediante el estándar OWL-S ampliado con protocolos FIPA.
- ❖ *ont* es la ontología que provee significado a todos los elementos del servicio. El lenguaje elegido es OWL-DL<sup>11</sup> (Web Ontology Language).

### **Organization Management System (OMS)**

El componente OMS es el principal responsable del manejo de las organizaciones y sus entidades. Es el responsable de los ciclos de vida de las organizaciones, incluyendo la especificación y administración de sus componentes estructurales (roles, unidades y normas) y sus componentes de ejecución (agentes participantes y roles que juegan, y unidades activas en cada momento).

- ❖ Un *rol* representa una posición dentro de la unidad donde se define. Lleva asociado unas reglas de interacción, impuestas por la estructura de su unidad y la posición concreta que ocupa en ella; y unas reglas de comportamiento, que especifican su funcionalidad (tipos de servicios que ofrece y requiere); limitan sus acciones (prohibiciones, obligaciones y permisos) y determinan las consecuencias de las mismas (sanciones y recompensas).
- ❖ Una *norma* indica las obligaciones, permisos y prohibiciones de los roles respecto al registro, solicitud y desempeño de servicios, su composición, o bien la calidad de sus resultados. Permite definir aquellas restricciones que no se pueden expresar en las precondiciones (o postcondiciones) de un servicio determinado.
- ❖ Una *unidad* representa a agrupaciones de agentes y admite la recursión (unidades dentro de otras). Establece así la estructura topológica del sistema. Por ejemplo, facilita la representación de estructuras de tipo jerárquico, matricial, coalición, etc.

---

<sup>11</sup> <http://www.w3.org/TR/owl-features/> [Último acceso 20/10/2009]

Para la gestión de estos componentes, el OMS maneja las siguientes listas:

- ❖ *UnitList*: guarda la relación de las unidades existentes, de sus unidades directamente superiores (SuperUnit), objetivos y tipo.
- ❖ *RoleList*: guarda la relación de los roles existentes en cada unidad, de qué roles heredan y cuáles son sus atributos (accesibilidad, posición).
- ❖ *NormList*: guarda la relación de las normas del sistema.
- ❖ *EntityPlayList*: guarda la relación de las unidades en las que se ha registrado como miembro cada agente, así como el rol que juega en dicha unidad.

El OMS ofrece los servicios necesarios para el adecuado funcionamiento de una organización. Dichos servicios se clasifican en: *servicios estructurales*, que modifican la especificación estructural y normativa propia de la organización; y *servicios dinámicos*, que permiten la gestión de la entrada y salida dinámica de agentes dentro de la organización, así como la adopción de roles.

Las organizaciones son estructuradas por medio de unidades, las cuales representan grupos de entidades (agentes u otras unidades), y que se relacionan porque tienen propósitos comunes. Las unidades tienen una topología interna (por ejemplo, jerárquica, de grupo o plana) que impone restricciones a las relaciones entre los agentes y al control (por ejemplo, supervisión o información entre las relaciones).

En THOMAS, una unidad “virtual” es definida para representar el “mundo” del sistema en el cual los agentes participan por defecto. El OMS crea organizaciones dentro de esta unidad “virtual” por medio del registro de unidades, que también pueden estar compuestas a su vez por más unidades.

A parte de esto, los roles se definen en cada unidad. Los roles representan toda la funcionalidad necesaria para lograr el objetivo de la unidad. También pueden tener asociadas normas para controlar las acciones propias de un rol (por ejemplo, qué servicios de agentes jugando ese rol pueden pedir, ofrecer o servir o qué permisos tiene para acceder a recursos, etc.). Como resultado, los agentes pueden dinámicamente adoptar roles dentro de las unidades, por lo que el OMS controla este proceso de adopción de roles y controla qué entidades juegan qué rol en cada momento.

### **Platform Kernel (PK)**

El componente PK mantiene básicamente el manejo de los servicios básicos en una plataforma multi-agente. Por tanto, está encargado de gestionar el ciclo de vida de los agentes presentes en las distintas organizaciones y permite además disponer de un canal de comunicaciones (incorporando diferentes mecanismos de transporte de mensajes) que facilite la interacción entre las diversas entidades. Por otro lado, el PK ofrece una conectividad segura y los mecanismos necesarios que permiten la inter-conectabilidad multi-dispositivo para

aplicaciones que así lo requieran. Los servicios ofrecidos son en la mayoría de los casos heredados de FIPA.

Los servicios de PK necesarios en la arquitectura THOMAS son clasificados en cuatro clases: (i) *Registration*: servicios necesarios para añadir, modificar y eliminar agentes nativos de la plataforma; (ii) *Discovery*: servicios que proporcionan la funcionalidad de obtención de información sobre los agentes nativos en la plataforma; (iii) *Management*: servicios para controlar el estado de activación de los agentes nativos de la plataforma; (iv) *Communication*: servicios para la comunicación de agentes dentro de la plataforma y fuera de ella.

Desde un punto de vista global, la arquitectura THOMAS, plantea una integración total de forma que los agentes puedan ofrecer e invocar servicios de forma transparente a otros agentes o entidades, así como qué entidades externas puedan interactuar con los agentes de la arquitectura mediante el uso de los servicios ofertados.

La siguiente tabla muestra un resumen de los diferentes servicios ofrecidos en THOMAS.

**Tabla 1. Resumen de los servicios ofrecidos en THOMAS**

<b>SF Services</b>		
<b>Type</b>	<b>SF Service</b>	<b>Description</b>
Registration	Register Profile	Creates a new service description
	RegisterProcess	Creates a particular implementation (process) for a service
	ModifyProfile	
	ModifyProcess	Modifies an existing service profile
	DeregisterProfile	Modifies an existing service process
DeregisterProcess		
		Removes a service description
		Removes a service process
Affordability	AddProvider	Adds a new provider to an existing service process
	RemoveProvider	Removes a provider from a service process
Discovery	SearchService	Searches for a service that satisfies the user requirements
	GetProfile	Gets the description (profile) of a specific a service
	GetProcess	Gets the implementation (process) of a specific a service
<b>OMS Services</b>		
<b>Type</b>	<b>OMS Service</b>	<b>Description</b>
Structural	RegisterRole	Creates a new role within a unit
	RegisterNorm	Includes a new norm within a unit
	RegisterUnit	

	DeregisterRole	Registration	Creates a new unit within a specific organization
	DeregisterNorm		Removes a specific role description from a unit
	DeregisterUnit		Removes a specific norm description
			Removes a unit from an organization
Information	InformAgentRole		Indicates roles adopted by an agent
	InformMembers		Indicates entities that are members of a specific unit
	QuantityMembers		Provides the number of current members of a specific unit
	InformUnit		Provides unit description
	InformUnitRoles		Provides unit description
	InformRoleProfiles		Indicates which roles are the ones defined within a specific unit
	Infor, RoleNorms		Indicates all profiles associated to a specific role
			Provides all norms addressed to a specific role
Dynamic	AcquireRole		Requests the adoption of a specific role within a unit
	LeaveRole		Requests to leave a role
	Expulse		Forces an agent to leave a specific role

### **Execution Framework**

El componente PK de THOMAS no solo controla las comunicaciones si no también el ciclo de vida de los agentes internos. Los servicios ofrecidos son heredados de FIPA, aunque con ciertos matices.

La novedad aquí está principalmente en los módulos OMS y SF, en la forma en que ellos son los que manejan y gestionan los sistemas abiertos. Además, pueden ser vistos como un todo, como un conjunto de servicios independiente de la plataforma en un marco para gestionar las organizaciones virtuales para sistemas abiertos. Esto lo que se conoce como *THOMAS Framework*.

Este "marco de ejecución" permite a cualquier agente (desde cualquier tipo de plataforma) crear su propia organización virtual con una estructura y unas normas que manejar, junto con la oferta y la demanda de servicios necesarios.

El *framework* manejará la estructura, las normas y el ciclo de vida de la organización, junto con la visibilidad de la oferta y demanda de servicios y el cumplimiento de las condiciones de uso.

Para ser lo más accesible posible, todas las funcionalidades del *framework* se ofrecen en dos formas diferentes: como un conjunto de servicios

independientes disponibles a través de descripciones WSDL<sup>12</sup> (Web Services Description Language), como servicios web comunes, que se pueden asignar en cualquier máquina o conjunto de ordenadores; o a través de mensajes ACL que permiten una comunicación de nivel superior o incluso procesos de negociación.

Desde el punto de vista de una organización virtual, todos los agentes incluidos en el *framework* deben pertenecer a una organización. Así pues, THOMAS *framework* ofrece una organización virtual en el que cualquier entidad puede incluirse automáticamente, así como una función general que permite a la entidad pedir descripciones de los servicios a fin de cumplir con sus necesidades. A través de la descripción del servicio, el cliente está informado de los roles necesarios para solicitar cualquier servicio específico, o de los roles necesarios para poder prestar un servicio específico dentro de la organización.

---

### 3.2 COORDINACIÓN

---

Cada organización necesita de un soporte de coordinación que determine explícitamente cómo deben organizarse y llevar cabo las acciones y tareas dentro de la misma. De manera general, en cualquier SMA, los sujetos cuyas actividades necesitan ser coordinadas son los agentes y las entidades entre las cuales surgen las dependencias suelen ser las metas, las acciones y los planes. Un *mecanismo de coordinación* determina el modo en el que uno o varios agentes realizan las tareas [Ossowski, 1998].

A lo largo de los años, se han propuesto mecanismos para manejar la coordinación en SMA de modos muy diferentes [Wooldridge y Jennings 1994]. Los más utilizados han sido los métodos "mediados". Al igual que ocurre en muchos ámbitos de las sociedades humanas, los agentes intermediarios juegan un papel importante en las sociedades artificiales. Estos agentes proporcionan medios para facilitar la coordinación entre agentes en entornos abiertos, mediante servicios de localización y comunicación entre agentes que proporcionan servicios, y agentes que los solicitan [Klusch y Sycara, 2001]. Por ejemplo, las primeras arquitecturas se basaban en el concepto de "mediador" [Wiederhold, 1992] como un modo de integrar recursos heterogéneos. Un ejemplo de arquitectura para SMA basado en el concepto de mediador es RETSINA (*Reusable Task Structure-based Intelligent Network Agents*). RETSINA fue desarrollada sobre la idea de que los agentes en el sistema forman una comunidad de iguales. La coordinación debería surgir de las relaciones entre los agentes en lugar de ser impuestas por la arquitectura, y como tal, no utiliza

---

<sup>12</sup> <http://www.w3.org/TR/wsdl> [Último acceso 20/10/2009]

un control centralizado sino que proporciona servicios (mediación) que facilitan las relaciones entre los agentes [Sycara et al., 2003].

En este trabajo, debido a la utilización de THOMAS, las organizaciones están compuestas por Unidades Organizativas, que proporcionan limitaciones comunicativas y de visibilidad a los agentes. Estas Unidades Organizativas pueden ser principalmente de tres tipos: *jerarquía*, *equipo* o *plana*. En las jerarquías, un agente supervisor tiene el control sobre el resto de miembros, coordina las tareas y centraliza la toma de decisiones. En los equipos, todos los miembros colaboran entre ellos para alcanzar un objetivo común, compartiendo su información, y la coordinación surge mediante planes y decisiones acordadas por los miembros. Por último, en las unidades planas no existe control de unos miembros sobre otros, de modo que los miembros pueden conocer la existencia del resto de miembros de la estructura. Esta última unidad es empleada principalmente para el modelado de estructuras más complejas. Usando este concepto de Unidad Organizativa se pueden definir estructuras organizacionales más elaboradas y complejas, como la estructura matricial, la federación, la coalición o la congregación que veíamos en la sección 3.1.

De forma general puede decirse que todas las arquitecturas tienen un punto en común: la preocupación por coordinar la conducta de los agentes. Los agentes pueden coordinar sus conocimientos, objetivos, habilidades, tareas y planes juntamente para tomar una acción o resolver una meta global. La coordinación permite a los agentes considerar todas las tareas a realizar y regularlas para no ejecutar acciones no deseables, como por ejemplo:

- ❖ Que los agentes no generen y comuniquen sub-soluciones que lleven al progreso en la solución de un problema.
- ❖ Que los agentes generen y comuniquen resultados redundantes.
- ❖ Que haya una distribución inapropiada de la carga de trabajo entre los agentes.

Desde una perspectiva práctica, es mejor, probablemente, concebir la coordinación como el esfuerzo de administrar el *espacio de interacción* [Wegner, 1997][Busi et al., 2001] de un SMA. Esta coordinación está relacionada con la planificación de acciones para la resolución de tareas, porque estos planes permiten:

- ❖ Conocer a alto nivel y predecir el comportamiento de otros agentes del sistema.
- ❖ Intercambiar resultados intermedios que lleven al progreso en la resolución de la tarea global.
- ❖ Evitar acciones redundantes, si ellas no son deseables.



La realización de las tareas por parte de los agentes, puede ser organizada de varias formas, por ejemplo: cada agente ejecuta una de las tareas, o bien, las tareas son divididas en sub-tareas, por medio de algún mecanismo de descomposición de problemas, y éstas son las realizadas por los agentes. En una organización, las tareas que debe realizar un agente dependen, entre otros factores, del rol que este agente asume en el sistema. Por ejemplo, en un sistema "oficina", un agente "persona" asume el rol de "secretaria" y realiza las labores relacionadas con ese rol; este mismo agente podría asumir el rol de "jefe" y realizaría labores muy diferentes, relacionadas con su nuevo rol. Para la realización de tareas, un agente puede necesitar recursos del sistema, en este caso tiene que coordinarse con los otros agentes del sistema que deseen usar el mismo recurso.

---

### 3.2.1 Modelos de coordinación

---

Como se puede ver, la coordinación en SMA ha sido, y es, un reto para los investigadores relacionados con la tecnología de agentes, existiendo diferentes aproximaciones. Existen planteamientos *bottom-up* ("de abajo a arriba"), como por ejemplo la gestión de objetivos cuando los miembros del grupo toman el control de la definición de su trabajo [Malone y Crowston, 1994]; y planteamientos *top-down* (de arriba a abajo), como por ejemplo ontologías compartidas [Fox y Gruniger, 1998] o la asignación jerárquica de responsabilidades utilizada en muchas organizaciones humanas.

Otros ejemplos de modelos de coordinación son los conjuntos de intenciones (*join-intentions*) [Cohen, Levesque, 1991], [Dunin-Keplicz, Verbrugge, 2002], los planes compartidos (*shared plans*) [Grosz, Kraus, 1996] y los modelos de equipo independientes del dominio [Tambe, 1997]. Todos los enfoques están basados en las observaciones de equipos de trabajo humanos. El enfoque de coordinación propuesto por la teoría de los planes compartidos es el que más se asemeja al modelo propuesto en este trabajo. La formalización de los planes compartidos enfatiza la necesidad de un modelo de equipo común y de alto nivel que permita a los agentes comprender todos los requisitos de los planes que podrían llevarse a cabo en el sistema, como objetivo de grupo. Esto permite a los miembros del equipo poner a disposición sus capacidades para llevar a cabo el plan y conseguir el objetivo global.

Como vemos, existen muchos *modelos de coordinación*, pero dos principales (que se mencionan a continuación) y gran cantidad de modelos intermedios.

- a) Coordinación Global. Cuando el SMA determina y planifica globalmente las acciones de los diferentes agentes.
- b) Coordinación Individual. Cuando el SMA le da completa autonomía a los agentes, es decir, cada agente decide qué hacer y resuelve localmente los conflictos que detecte con otros agentes.

La *Coordinación Global* está vista desde una perspectiva *macro* (centrada en el SMA) y en ella las consecuencias de la coordinación pueden ser concebidas como un plan (o decisión, acuerdo, acción, etc.) "global". Este plan puede ser un "plan compartido" [Rosenschein y Zlotkin, 1994][Grosz, Kraus, 1996] si los agentes llegan a un acuerdo explícito durante el proceso de coordinación, o puede ser simplemente la suma de los planes (o decisiones, acciones, etc.) individuales de cada agente (lo que algunas veces se denomina "multiplan" [Ossowski, 1998]).

La *Coordinación Individual* está vista desde una perspectiva *micro* (centrada en el agente). [Von Martial, 1992] entiende la coordinación como una forma de *adaptación al entorno*. El agente de manera individual se coordina con el resto reconsiderando sus metas, planes y acciones iniciales. A este nivel, para el agente, el resultado de la coordinación (es decir, la adaptación individual) es mejor cuanto más contribuya a la consecución de sus meta en el nuevo entorno multi-agente.

Cuando centramos la atención en el nivel micro, entra en juego la distinción entre modelos *cuantitativos* y *cualitativos* [Ossowski, 1998]. El enfoque cualitativo representa de forma directa las diferentes razones por las cuales se prefieren ciertos objetos de coordinación a otros. El comportamiento de coordinación local de un agente se basa en si sus acciones (planes, metas, etc.) dependen positiva o negativamente de las acciones de los demás: elegirá sus acciones locales basándose en una "estructura" de dependencias que comparte con los demás. Así, concibe la coordinación como un *problema de satisfacción de restricciones* [Ossowski, 2001]. En los modelos cuantitativos, al contrario, la estructura del problema de la coordinación se esconde bajo la forma de una función de utilidad multi-atributo. El problema de decisión acerca de su coordinación local equivale entonces a una clase especial de problema de optimización: determinar una acción local (plan, meta, etc.) e inducir al resto a elegir acciones locales (planes, metas, etc.) para así maximizar su utilidad local [Ossowski, 1998]. La aproximación cuantitativa tiene la ventaja de que puede hacer uso de un marco fundamentado tanto en el caso de escenarios cooperativos (Investigación Operativa) como en el caso de escenarios no cooperativos (Teoría de Juegos), pero la desventaja es que, debido a las incertidumbres intrínsecas de los dominios de los SMA, la función de utilidad es solamente una aproximación, de modo que su óptimo no coincide necesariamente con la mejor opción de un agente en el entorno real. El modelo cualitativo es menos propenso a esto, pero su base son teorías sociales que no proporcionan un marco formal que guíe la toma de decisiones.

Entre los modelos intermedios podemos destacar otras dos dimensiones: *coordinación directa* o *mediada*, y *orientada por problemas* u *orientada por cooperación*:

- c) En la coordinación *directa*, las interacciones se realizan de modo *peer-to-peer* ("entre iguales"), involucrando únicamente los agentes que deseen coordinarse.
- d) La coordinación *mediada* se basa en la existencia de agentes intermediarios que ofrecen servicios entre los agentes.

La coordinación directa suele implementarse de forma descentralizada y es más apta para entornos abiertos en los que se realiza una coordinación en tiempo de ejecución. En cambio, tradicionalmente la coordinación mediada se realiza de forma centralizada (mediante "agentes coordinadores") y en entornos cerrados que permiten establecer los mecanismos de coordinación en tiempo de diseño. Sin embargo, actualmente se tiende a instrumentar una coordinación mediada de modo "multi-céntrico", en parte centralizado y en parte descentralizado, en base a agentes intermediarios múltiples y heterogéneos. De este modo, se pretende atender tanto a las necesidades ingenieriles de eficiencia como a las restricciones impuestas por entornos cada vez más abiertos. Tómese como ejemplo, el modelo propuesto en esta tesis.

- e) Coordinación *orientada por los problemas*: en este tipo de coordinación, los agentes deben coordinar los planes de realización de acciones para prevenir interbloqueos, repetición de acciones y creación de inconsistencias.
- f) Coordinación *orientada por la cooperación*: en este tipo de coordinación los agentes no se coordinan a nivel de planes, sino a nivel de acciones. Esto significa que los agentes se coordinan en el momento de ejecutar acciones.

La elección del modelo de coordinación dependerá del ámbito de actuación de la propia organización, pero siempre debe intentarse cubrir todas las posibilidades de actuación de los agentes. En este sentido, el modelo de coordinación para organizaciones propuesto en este estudio cubre todos los tipos de coordinación mostrados ya que la organización determinará globalmente las acciones de los agentes (coordinación global) pero dejando libertad a cada uno de ellos para que decida cómo resolver sus propios planes (coordinación individual). Además, la coordinación de la organización (a nivel *macro*) estará *orientada a problemas*, previniendo inconsistencias entre planes de cada agente y haciendo que cada uno se coordine a nivel de acciones cuando de manera global se decida el plan de la organización (a nivel *micro*).

Dentro del enfoque social todavía se presentan limitaciones en el sentido de que los estudios sobre la coordinación están principalmente centrados en las acciones dentro del sistema para cada agente. Es decir, no consideran las motivaciones generales o globales de la organización.

En este estudio se presenta una aproximación para llevar a cabo una coordinación global dentro de una organización desarrollada mediante THOMAS; que como hemos visto, es una arquitectura basada en conceptos organizacionales. La coordinación consistirá en la planificación de la distribución de las tareas de los agentes miembro de la organización. Además, esta coordinación podrá adaptarse a cambios en los planes concebidos en la organización, dando a la misma la propiedad "adaptativa". Una descripción de las principales tendencias de esta propiedad en las organizaciones es dada en el siguiente apartado.

---

### 3.3 ADAPTACIÓN

---

Existen algunos problemas que deben tenerse en cuenta a la hora de coordinar los agentes de una organización, como poder mantener la coherencia global del sistema sin tener un control global explícito [Huhns y Stephens, 1999]. Para que una organización pueda adaptarse rápidamente a los cambios de su entorno, los agentes deben poder coordinarse cuando es necesario realizar cambios en sus metas o los roles que tienen asignados. Es decir, es necesario que *la organización se adapte*.

Esta afirmación, transforma la tarea de coordinación de los agentes en un aspecto fundamental de cualquier SMA [Doran et al., 1997]. La coordinación puede darse entre un conjunto de agentes que cooperan para realizar una tarea conjunta (coordinación global), o bien, entre un conjunto de agentes individualistas o competitivos que persiguen sus metas particulares y deben negociar por recursos compartidos o que son brindados por otros agentes del conjunto (coordinación individual). En el caso de un sistema cooperativo (coordinación orientada por cooperación), la coordinación consistirá en la planificación de la distribución de las tareas, mientras que en el caso competitivo (coordinación orientada por problema) la coordinación consistirá de una negociación entre los agentes [Jennings et al., 1998] [Rueda et al., 2002].

Por lo tanto, una organización virtual puede verse como un sistema cooperativo, en el que la coordinación se basa en una planificación y distribución de tareas. La coordinación de tareas compartidas o tareas que se combinan para resolver un problema común, requerirá o bien de una planificación centralizada, o bien de una planificación distribuida llevada a cabo por los propios agentes del sistema. Esto constituye un problema abierto, ya que los sistemas clásicos de planificación [Sacerdoti, 1975][McAllester y Rosenblitt, 1991][Penberthy y Weld, 1992] no son apropiados por varios motivos:

1. Asumen que los agentes tienen un conocimiento completo de su entorno.

2. Asumen que las acciones no van a fallar.
3. Asumen que el entorno cambiará únicamente por la ejecución de las acciones del agente.

Ninguna de estas tres suposiciones puede realizarse de forma realista en un SMA, por lo tanto se requiere de una planificación que pueda adaptarse a estas circunstancias.

El modelo propuesto pretende llevar a cabo una planificación adaptativa dentro de una organización de agentes que no asuma estas suposiciones y pueda aplicarse a entornos reales. Como vimos en el apartado 2.1.5, existen diferentes paradigmas de planificación, pero el más adecuado para llevar a cabo este objetivo es la planificación basada en casos ya que nos proporciona la flexibilidad necesaria para llevar a cabo una replanificación de las tareas, acciones y objetivos de la organización. Esta flexibilidad viene dada por la posibilidad de utilizar planes o sub-planes basados en experiencias pasadas. Si además, este tipo de planificación se combina con una planificación de satisfactibilidad, es posible introducir restricciones temporales y de recursos en el problema. El conjunto de planes obtenido es compatible con las restricciones impuestas en el sistema y permiten a un agente pasar de un estado inicial a un estado objetivo manteniendo la eficiencia constante.

La planificación basada en casos permite la adaptación del sistema desde el punto de vista de la planificación dinámica. Es decir, permite formalizar los conceptos de *replanificación* y de plan *fácilmente replanificable*, mediante técnicas basadas en campos de Jacobi [Lee, 1997]. Si tenemos un conjunto de planes con eficiencia constante, mediante estas técnicas es posible determinar cuál de ellos presenta mayor densidad de planes alternativos en su entorno y, en el caso de ser interrumpido, cuenta con un mayor número de planes próximos con los que continuar. Como resultado, se obtiene el plan que más fácilmente permitiría a un agente adaptarse. Se le da al agente la habilidad de poder reaccionar dinámicamente frente a situaciones inesperadas y tener su propio plan. Pero además, es necesario tener en cuenta que también debe ser capaz de coordinarse con el resto de agentes de la organización, de tener una perspectiva de grupo que no solamente genere la mejor opción para él, sino para toda la sociedad del que es miembro.

Siguiendo este orden de ideas se llega a preguntas tales como qué se debe adaptar, cómo se debe adaptar y de quién es la responsabilidad de la adaptación. Dichas cuestiones finalmente están determinadas por el objetivo que la organización debe o quiere cumplir, factor que se ve influenciado ampliamente por la convivencia en sociedad. De esta manera, las funciones de utilidad o eficiencia que determinan qué decisiones tomar pueden estar dadas por un beneficio individual o colectivo.

La complejidad de este problema va atada al hecho de no tener reglas generales acerca de la adaptación, en donde cada uno de los métodos existentes funciona adecuadamente para situaciones determinadas. Específicamente en el problema de adaptación social, la situación es aún más complicada, ya que factores inherentes al sistema cambian por completo los paradigmas de adaptación y donde los aspectos tenidos en cuenta inicialmente para un individuo dejan de ser trascendentales para pasar a un segundo plano determinado por el comportamiento del grupo, cambiando lo que en algún momento fue la estructura y funcionamiento de la adaptación para un individuo específico.

El modelo de adaptación de la organización propuesto utiliza mecanismos basados en interacciones directas [Zambonelli et al. 2004]: interacciones y cálculos locales para reorganizar los agentes y así obtener un estado global coherente del sistema (sección 2.2.2). Además, la utilización de la plataforma THOMAS [Carrascosa et al., 2009] [Giret et al., 2009], nos acerca a nuevos métodos de adaptación basados en arquitecturas [Razavi et al., 2005] en los que es posible modificar la estructura de la organización para adaptarse a los cambios.

En general vemos que se pueden distinguir diferentes enfoques organizacionales: MOISE [Boissier 2007], OperA [Dignum 2004], INGENIAS [Pavon 2003], the Gaia Methodology [Zambonelli et al., 2003], [Omicini et al., 04]. Pero mientras estos trabajos proporcionan mecanismos para la coordinación entre participantes, existe mucho menos trabajo centrado en la adaptación en tiempo de ejecución de las estructuras organizacionales o normas definidas en el diseño.

Por ejemplo [Hubner, 2004] propone un modelo para controlar la adaptación creando nuevas normas. [Gasser y Ishida 1991] proponen un modelo distribuido para reorganizar su arquitectura. [Artikis et al., 2009] hace que los agentes sigan un protocolo para adaptar sus normas. Todos estos trabajos están **centrados en la estructura**, basando su adaptación en la coordinación entre participantes.

Existen otros enfoques como los **basados en normas sociales** (en el rol del grupo social y efecto en la aparición y apoyo de normas sociales), como por ejemplo [Hales 2002] o [Lakkaraju y Gasser, 2008].

Otra posibilidad es el desarrollo de SMA con un enfoque basado en el concepto de **organización/institución**. Las interacciones humanas están reguladas por instituciones [North, 1990] que representan las reglas del juego en una sociedad y definen lo que los individuos pueden y no pueden hacer y bajo qué condiciones. Esta perspectiva defiende la adopción de una estrategia mimética con el objeto de afrontar la complejidad del desarrollo de SMA abiertos bajo un concepto organizacional. Una institución electrónica [Noriega et al., 1999][

Rodríguez-Aguilar, 2003][Esteva, 2003][García et al., 2009] debe ser considerada como un middleware social entre los agentes externos participantes y la capa de comunicación escogida encargada de aceptar o rechazar las acciones de los agentes. También existen trabajos que direccionan este tema a la definición de "Automatic Electronic Institutions", una extensión de las instituciones electrónicas dotadas con capacidades de auto-adaptación [Bou et al., 2007a][Bou et al., 2007b][Bou et al., 2009]. La principal diferencia con las otras propuestas es que la adaptación es llevada a cabo por la institución, en lugar de por los agentes.

Otros puntos de vista [Shoham, 1992][Shoham y Tennenholtz, 1997][Kittock, 1993] [Walker, 1995] se centran en **mecanismos de regulación de comportamientos**. Aquí el factor clave es la "imitación", es decir, las convenciones sociales ocurren a través de propagación y contagio [Conte y Paolucci, 2001][Burt, 1987].

Por último existen enfoques en los que los agentes utilizan la información obtenida durante sus interacciones [Axelrod 1986][Castelfranchi y Conte, 1995][Sen y Airiau, 2007]. En estos trabajos las normas sociales (como tipos especiales de convenciones sociales) son usadas por los agentes para auto-regular su comportamiento según la **información social**: historia previa, reputación, etc. La investigación se centra en mecanismos de grupos sociales basados en la información social recopilada durante las interacciones [Villatoro y Sabater-Mir, 2008a][Villatoro y Sabater-Mir,2008b][Villatoro y Sabater-Mir,2008c]. Investigan los efectos de los algoritmos de transmisión de información para reconocer y formar grupos sociales.

En ninguno de los enfoques estudiados, se lleva a cabo una coordinación de tareas de los agentes miembros de la organización que se combinen para resolver un problema común, y que además, tengan en cuenta que la planificación de las tareas debe adaptarse a los cambios del entorno. El modelo social utilizado en la arquitectura elegida, THOMAS [Carrascosa et al., 2009] [Giret et al., 2009], está centrado en la definición de la estructura y las normas. La adaptación en este tipo de modelos se basa en la coordinación entre los participantes de la organización.

Aprovechando las ventajas que nos proporcionan las particularidades del desarrollo de SMA desde el punto de vista organizacional, y considerando las carencias existentes actualmente respecto a una planificación adaptativa para un modelo social cualquiera, se propone en este trabajo un modelo de coordinación para la planificación dinámica y adaptativa en una organización de agentes. Mediante la combinación de tecnología multi-agente y de técnicas de planificación basadas en investigación operativa, se pretende llevar a cabo una distribución óptima de las tareas de los agentes miembros de la organización. Se trata de un modelo único en su concepción, que permite dotar a una organización de capacidades auto-adaptativas en tiempo de ejecución

para entornos altamente dinámicos. De esta forma, el comportamiento de un agente estará determinado por las metas que desee alcanzar, pero teniendo en cuenta el resto de metas de los agentes compañeros y los cambios en el entorno.



---

## 4 TECNOLOGÍAS COMPLEMENTARIAS

---

Los agentes y los sistemas multi-agente cuentan con características importantes, entre las que destacan la autonomía, la reactividad, la proactividad, el aprendizaje, así como una comunicación ubicua, distribuida, y en cierto modo inteligente entre sus elementos. Haciendo uso de estas características, es posible cubrir gran parte de los requerimientos que plantean los problemas de coordinación y planificación ya vistos, especialmente en el diseño de sistemas capaces de adaptarse a los cambios en el entorno de forma ubicua, autónoma y dinámica.

El grupo de investigación BISITE (Biomedicina, Sistemas Inteligentes y Tecnología Educativa) de la Universidad de Salamanca, cuenta con una amplia experiencia en las áreas de la Inteligencia Artificial y el desarrollo de sistemas multi-agente. Por tal motivo, esta tesis doctoral sigue una línea continuista en la investigación y aplicación de estos sistemas, extendiendo sus posibilidades hacia otras áreas, en especial hacia las organizaciones de agentes.

Además de profundizar en la tecnología de agentes, es importante describir los sistemas de razonamiento y planificación que se han utilizado para expandir las capacidades de estos sistemas. Por otra parte, en la actualidad se está abriendo una nueva línea de investigación respecto a la simulación, representación y estudio del comportamiento social de los agentes inteligentes, los cuales son representados en modelos tridimensionales en entornos capaces de interactuar con los usuarios. Respecto a este último campo, en este apartado se describirán los principales conceptos de los agentes virtuales inteligentes.

---

### 4.1 SISTEMAS CBR Y CBP

---

El razonamiento basado en casos (CBR) es un tipo de razonamiento, utilizado en el pensamiento humano, en el que se recurre a experiencias pasadas para resolver nuevos problemas [López de Mántaras y Plaza, 1997]. Así pues, si en un momento pasado se decidió resolver un problema utilizando una determinada solución (o grupo de soluciones) y, una vez aplicada esa solución se obtuvo un determinado resultado, entonces parece lógico que si se presenta un nuevo problema con características similares al que se resolvió previamente en el pasado, se recurra a la experiencia adquirida en la experiencia pasada para dar solución al nuevo problema. De esta forma, sabemos que disponemos en la memoria de la experiencia correspondiente a la resolución de un problema similar al actual. Si el resultado obtenido tras aplicar la solución en el pasado fue bueno, entonces parece lógico aplicar ahora una solución similar

para resolver el problema actual. Por el contrario, si el resultado que se obtuvo en el pasado no fue bueno, entonces la opción lógica debe ser la de modificar la solución que se aplicó en el pasado buscando obtener un resultado mejor. Se trata pues, de razonar a partir de recuerdos. Una vez finalizado el proceso, la persona ha realizado un aprendizaje, ya que ha aprendido de la nueva experiencia vivida. Hay que tener en cuenta dos aspectos del modelo de razonamiento basado en la experiencia. En primer lugar, el modelo se basa en la idea de que problemas similares tienen soluciones similares. Sin embargo, carecer de problemas similares no supone que el sistema no sea capaz de proponer buenos resultados, sino que la reutilización de memorias pasadas se convierte entonces en un proceso creativo. Sea cual sea el resultado de este proceso creativo, el individuo aprende de la nueva experiencia. En segundo lugar, en el proceso de razonamiento explicado se está emitiendo un juicio de valor que permite saber si la solución aplicada para resolver un determinado problema fue buena o no fue buena. Si este juicio es emitido por un experto en la materia del problema resuelto, mayores serán las posibilidades de incrementar la capacidad de aprendizaje [Schank, 1982].

El término CBR es muy general, y existen diferentes tipos o especializaciones para el razonamiento basado en casos, especialmente atendiendo a la representación, indexación o a los mecanismos de razonamiento aplicados sobre los casos [Aamodt y Plaza, 1994]. Típicamente un sistema CBR propiamente dicho viene caracterizado por el concepto de caso. Un caso debe contener cierto nivel de información y cierta complejidad en su estructura interna (con lo que, por ejemplo, un simple vector no sería considerado un caso). Además, este tipo de sistemas son capaces de adaptarse a distintos entornos o contextos, facilitando la generalización y cierto grado de independencia del entorno [López de Mántaras, 2001].

Cuando las soluciones que se proponen son planes, entonces se habla de una especialización del CBR denominada planificación basada en casos (CBP). En la mente humana, el concepto de intención se encuentra íntimamente ligado al concepto de plan. Los planes guían las acciones humanas. [Hammond, 1989] propuso la planificación basada en casos, entendiendo la planificación como una tarea de la memoria. De esta forma se establece una analogía entre la planificación y el recuerdo. Se podrán reutilizar, descartar o volver a aplicar recuerdos de aciertos, fallos o modificaciones realizadas en el pasado. Según Hammond, un sistema CBP utiliza el conocimiento que posee de su entorno y los efectos que tienen sus acciones en su entorno para construir planes. Así el plan construido sirve para testear dicho conocimiento. Las ventajas que presenta el CBP con respecto a otras aproximaciones a la planificación son: CBP permite anticiparse a posibles problemas que puedan aparecer durante la ejecución del plan y descartar aquellos planes que puedan presentar problemas. También permite la depuración de aquellos planes fallidos. Finalmente, CBP permite almacenar planes para su uso posterior.

Hasta ahora se ha hablado de razonamiento basado en casos y planificación basada en casos. Se ha explicado que se trata de un razonamiento o planificación basados en la experiencia, en recuerdos. Probablemente haya llamado la atención que en los nombres que se dan tanto al razonamiento como a la planificación se utilice la palabra casos. El concepto de caso es fundamental en los sistemas CBR/CBP, tanto que la estructura de un sistema CBR/CBP se diseña sobre el concepto de caso. Para [Schank, 1982] un caso es un pedazo de conocimiento contextualizado que representa una experiencia, de tal forma que un caso contiene:

- ❖ el problema que describe el estado del mundo cuando ocurrió el caso,
- ❖ una descripción de la solución encontrada y/o
- ❖ un resultado que describe el estado del mundo después de que ocurrió el caso.

De esta forma, un caso se puede ver como una lección aprendida cuando se ha resuelto un determinado problema. Los casos pueden representarse de diversas formas, como instancias, como vectores, como objetos, etc. Un caso se representa por medio de características llamadas índices. Los índices podrán ser restricciones, objetivos, solución, fracasos, etc. Según [Kolodner, 1993] un caso se compone de una descripción del problema, de la solución para ese problema y del resultado obtenido tras aplicar la solución. De esta forma se puede utilizar una notación matemática en la que un caso se represente por medio de una 3-tupla,  $\langle P, S(P), R \rangle$ , y es la que se muestra a continuación.

**Case: <Problem, Solution, Result>**

Problem: initial\_state

Solution: sequence of <action, [intermediate\_state]>

Result: final\_state

Así pues, un caso queda definido a través de tres elementos:

- ❖ P o Problema: Representa una descripción del problema que se desea resolver.
- ❖ S(P) o Solución: Es la solución del problema P, y viene representada a partir del conjunto de operadores que se utilizan para construir la solución.
- ❖ R o Resultado: Es la eficiencia, que mide los recursos utilizados para alcanzar la solución.

[Riesbeck y Schank, 1989] proponen una estructura que permite modelar un sistema CBR, de tal forma que se describen los componentes principales del sistema y las relaciones que se establecen entre ellos. El sistema debe estar formado por dos componentes fundamentales: la memoria de casos y el mecanismo de razonamiento (que a su vez puede necesitar de la inclusión de una base de conocimiento).

- ❖ El **mecanismo de razonamiento** o modelo de actuación de un sistema CBR se conoce como ciclo de vida de un sistema CBR. En el ciclo

de vida de un sistema CBR se especifican los pasos ordenados y relacionados por tiempo mediante los cuales se extrae y aprende información para resolver un problema específico. El ciclo de vida de un sistema CBR está formado por cuatro procesos secuenciales [Aamodt y Plaza, 1994]. Estos cuatro procesos son conocidos como las cuatro "erres": "*Retrieve*" (Recuperación), "*Reuse*" (Adaptación), "*Revise*" (Revisión) y "*Retain*" (Retención o aprendizaje) [López de Mántaras *et al.*, 2005].

- Etapa "*Retrieve*" (Recuperación). Es la primera etapa que realiza un sistema CBR. En esta etapa se realiza la recuperación de casos, esto es, el acceso a los casos almacenados que cuentan con una descripción de problema más similar a la del problema actual. En esta etapa se llevan a cabo dos funciones distintas: Acceso a los casos almacenados y establecer la similitud entre casos (más concretamente entre descripciones de problema). De esta forma es necesario plantearse:
  - Algoritmo de acceso a los casos almacenados: Preferiblemente debe tratarse de un algoritmo que garantice un acceso rápido y eficiente.
  - Técnicas o métricas que permitan determinar la similitud entre casos: A aquellos casos a los que se accede mediante el algoritmo de acceso se les aplica una métrica de similitud que permita determinar cuál o cuáles de ellos son los mejores casos (los más similares al problema actual) [Golding y Rosenbloom, 1988].
- Etapa "*Reuse*" (Reutilización o Adaptación). Una vez finalizada la etapa de recuperación, el sistema CBR pasa a ejecutar la etapa de reutilización. Esta etapa recibe como entradas los casos más similares recuperados durante la etapa anterior. La reutilización o adaptación consiste en trabajar con las soluciones correspondientes a los casos más similares recuperados en la etapa anterior para poder obtener una solución para el problema actual. Trabajar con las soluciones significa modificarlas y combinarlas, o simplemente decidir cuál de ellas es la más óptima y reutilizarla. Al igual que ocurría en la etapa anterior existen muchas posibilidades para realizar la adaptación. Lo ideal es encontrar un caso con una descripción de problema idéntica al actual. Pero, dado que en el mundo real es muy complicado que se presenten dos situaciones idénticas, se hace necesario crear una nueva solución basándose en las soluciones similares de las que se dispone. Además de los datos proporcionados por los casos similares recuperados se hace necesario utilizar algún tipo de conocimiento. El conocimiento puede venir dado a través de fórmulas o reglas.

- Etapa “*Revise*” (Revisión). Al igual que ocurre con la etapa de recuperación, el proceso de evaluación de la bondad de la solución finalmente aplicada para resolver el problema actual puede ser problemático, puede ser necesario de un conocimiento considerable [Leake et al., 2000]. Se comprueba si la solución propuesta en la etapa anterior es apropiada para el caso actual. Para ello se utiliza un sistema experto de conocimiento, o bien una persona experta. El resultado de esta etapa es un nuevo caso, para el que se haya obtenido una solución satisfactoria o una solución incorrecta y deba ser reparado. En ocasiones, en esta etapa se puede realizar una reparación de los fallos o errores detectados.
- Etapa “*Retain*” (Retención y Aprendizaje). En esta última etapa se aprende a partir de la nueva experiencia adquirida. Para ello se almacena el caso actual y la solución aplicada para resolverlo. Además se tiene en cuenta el resultado obtenido en la etapa de revisión para asignar una eficiencia al caso. De esta forma el caso puede ser indexado en la memoria de casos. Puede ser necesario reorganizar la memoria de casos. En caso de similitud con otros casos se puede aplicar generalización. En ocasiones durante esta etapa, o bien como una etapa adicional previa a la retención y aprendizaje se incluye una revisión de la base de conocimiento. En el caso de que se estén utilizando reglas u otro tipo de conocimiento experto, el conocimiento es revisado y actualizado en función de los resultados obtenidos para la experiencia que se acaba de vivir.

❖ La **base de casos o memoria de casos** es una de las partes más importantes de la estructura de un sistema CBR. De ella se extraen las soluciones anteriores y en ella se almacena lo aprendido. La memoria de casos es la encargada de mantener la representación y organización de los casos. La base de casos debe tener en cuenta la estructura de los casos (representación de los casos), y debe tratar de facilitar en la mayor medida de lo posible cada una de las operaciones que se realizan en el ciclo CBR. Para facilitar dichas operaciones es necesario contemplar tanto la indexación y organización de los casos (asignación de índices para facilitar su recuperación) como el mantenimiento [Leake *et al.*, 2000]. Los modelos actuales tienden a incorporar tendencias como: (i) la de la representación mediante objetos [Bergmann y Stahl, 1998]; para Bergmann [Bergmann *et al.*, 2005] se tiene representación orientada a objetos, utilizando los conceptos de clases y objetos; (ii) representación mediante XML (CBML), un estándar que permite la representación de casos utilizando XML; (iii) representación textual, necesaria cuando los datos con los que se trabaja son textuales, es decir, se trabaja con palabras o frases; (iv) representación jerárquica, la cual tiene en cuenta distintos niveles de abstracción para representar un caso; (v) representación en aplicaciones para procesamiento de imágenes, una

representación especial para los casos en la que la definición del problema de la interpretación de imágenes incluye trabajar con informaciones indirectas (tipo y parámetros del sensor, luminosidad, información sobre objetos, etc.); y (vi), la representación en CBP, que se describirá en la siguiente subsección 4.1.1.

Adicionalmente a la memoria de casos puede hacerse necesaria la utilización de una **base de conocimiento**. La base de conocimiento contendrá informaciones tales como teorías, principios, reglas, etc. que permitan tomar decisiones para la solución de problemas y aprender de las experiencias. Tal y como se ha indicado, estos conocimientos suelen ser necesarios durante las etapas de adaptación y de revisión. Se trata de conocimientos que permiten realizar funciones tales como la generalización, la toma de decisiones, etc. Se trata de funciones complicadas en las que el mantenimiento también juega un papel muy importante. El conocimiento del que dispone el sistema puede ser un sistema experto. El sistema de conocimiento experto debe ser capaz de aprender de las experiencias adquiridas por el sistema CBR. Probablemente se trata de la parte más complicada de un sistema CBR.

A continuación se detalla el mecanismo de planificación basada en casos, que permite aprender a partir de los planes ejecutados en el pasado.

---

#### 4.1.1 *Planificación basada en casos*

---

La planificación basada en casos es una especialización del razonamiento basado en casos que requiere una representación más específica del concepto de caso [Cox *et al.*, 2006]. La razón es que es necesario representar planes como solución. Tal y como se ha indicado con anterioridad un plan es una secuencia ordenada de acciones que permite llegar desde un estado inicial a un estado final. La representación de un caso vendrá dada por la descripción del problema (estado inicial), la solución (conjunto de acciones ejecutadas) y el resultado obtenido tras aplicar la solución [Bergmann *et al.*, 2005]. El gran inconveniente que presenta esta representación es la notación utilizada para representar la solución. La solución puede verse como una secuencia de estados intermedios por los que se pasa para llegar desde el estado inicial hasta el estado final. Los estados se suelen representar como conjuntos de lógica proposicional. El conjunto de acciones suele representarse como el conjunto de operadores junto con la relación que indica su ordenación. Para [Carbonell, 1986] es necesario además añadir información adicional sobre las decisiones tomadas a lo largo del plan.

En la planificación basada en casos, un nuevo plan se realiza a partir de planes previamente generados [Hammond, 1989], [Veloso, 1994], [Cox *et al.*, 2006]. En la práctica, la información que se almacena para un plan determinado no es solamente la descripción del problema y su solución, sino también información adicional acerca de cómo se ha obtenido el plan, de las decisiones que se toman,

de por qué el plan funciona o por qué no funciona, etc. Los diferentes planificadores basados en casos difieren unos de otros en la forma en la que representan y almacenan los casos y en la forma en la que ejecutan el ciclo CBP (en los algoritmos ejecutados en cada una de sus etapas [Bergmann *et al.*, 2005]).

La descripción formal de un planificador basado en casos [Yang, 1997], [Spalazzi, 2001], se lleva a cabo teniendo en cuenta la perspectiva de la notación utilizada en los modelos de planificación, un problema de planificación se plantea como una tupla de tres elementos  $\langle I, G, Op \rangle$ :

- ❖ I es un conjunto de fórmulas que permite describir el estado inicial.
- ❖ G es un conjunto de fórmulas que permite describir el estado o estados meta.
- ❖ Op es un conjunto de operadores (acciones) que pueden ser utilizadas por un plan. Cada acción  $a \in Op$  se suele describir en términos de sus pre-condiciones  $Ca$  (aquello que es necesario para que se pueda ejecutar la acción) y post-condiciones o efectos  $Ea$  (aquello que debe cumplirse cuando termine de ejecutarse la acción).

Un plan P es una tupla  $\langle S, B, O, L \rangle$  en la que:

- ❖ S es el conjunto de acciones que componen el plan. En este conjunto de acciones es necesario distinguir dos tipos especiales de acciones:
  - tI son aquellas acciones cuyos efectos son I, es decir, el estado inicial,
  - tG son aquellas acciones cuyas pre-condiciones son G, es decir, permiten pasar a un estado meta.
- ❖ O es una relación de ordenación sobre S que permite establecer un orden entre las acciones que componen el plan. Hay que hacer notar que tI es siempre la primera acción y que tG es la última acción. Cuando la relación de orden es total, P es un plan lineal, mientras que si la relación de orden es parcial, P es un plan no lineal.
- ❖ B es un conjunto que permite describir cuales son las vinculaciones permitidas y las vinculaciones prohibidos entre las variables que aparecen en P.
- ❖ L es un conjunto de enlaces causales de la forma  $s \rightarrow p \rightarrow s'$ , donde  $s, s' \in S$ ,  $p \in Es$  y  $p \in Cs'$ . Se trata de relaciones que permiten enlazar unas acciones con otras dentro del conjunto de acciones del plan.

Un plan P constituye la solución proporcionada para un problema de planificación cuando por cada acción  $s \neq tI$ , por cada  $p \in Cs$  existe un enlace causal  $s \rightarrow p \rightarrow s'$  y por cada acción  $s \neq tG$  existe al menos un enlace causal  $s \rightarrow q \rightarrow s''$ . En el caso de que se tengan en cuenta fallos o situaciones inesperadas

durante la ejecución de un plan, se representa como F. La representación de un plan es muy dependiente del modelo que se cree para el dominio.

La planificación basada en casos es una especialización del razonamiento basado en casos, en la que las soluciones que se generan son planes. Al igual que en un sistema CBR, los conceptos fundamentales en un sistema CBP son la representación que se utilice para la memoria de planes y la implementación elegida para las etapas del ciclo CBP. Se trata de factores que influyen de forma decisiva en la eficiencia del planificador. Al igual que ocurría en los sistemas CBR existen distintas posibilidades tanto para la representación de la memoria de planes como para la implementación de cada una de las cuatro etapas del ciclo CBP.

Para Spalazzi [Spalazzi, 2001] las posibles opciones para la representación de la memoria de planes son:

- ❖ Caracterización pura o representación relacional.  
Se trata de utilizar una representación en la que un caso se compone de las características que describen el problema y de las características que describen la solución. Las características son atributos simples. Así la descripción del problema vendrá dada por medio de la situación inicial (I), las metas (G) y los posibles fallos (F). La solución se representa como un plan (P).

Sin embargo, para muchos autores este tipo de representación no es suficiente. En una representación relacional se tienen en cuenta las relaciones que pueden existir entre una característica y otras características de diferentes casos (relaciones is-a, part-of, etc.).

- ❖ Representación transformacional o derivacional.
  - La representación transformacional representa un plan con todas las acciones planeadas y con todas las submetas.
  - La representación derivacional describe una solución por medio de las decisiones tomadas a la hora de generar el plan (sub-metas consideradas, alternativas generadas, situaciones erróneas, etc.).
- ❖ Representación indexada o Flat.  
Se trata de los métodos más utilizados. Se utiliza un esquema de indexación que permite organizar la memoria en jerarquías.
- ❖ Representación basada en lógica.  
Se utiliza una terminología lógica o una lógica de características.

En cuanto a las etapas que forman el ciclo CBP, son muy similares a las de los sistemas CBR. Las etapas de recuperación y aprendizaje van a ser muy dependientes de la representación y organización elegidas para la memoria de planes. Quizá la etapa más interesante a tener en cuenta y que requiere una atención especial es la etapa de adaptación. En la etapa de adaptación es



necesario decidir la técnica que permita reutilizar los planes pasados (también será necesario considerar la representación y organización de la memoria de planes). Las principales opciones utilizadas como técnicas de adaptación son:

- ❖ Reutilización directa de alguno de los planes recuperados, de aquel que se considere óptimo.
- ❖ Reutilización manual, en la que el usuario adapta manualmente los planes recuperados.
- ❖ Satisfacción de restricciones, que se utiliza cuando no se desea cambiar la estructura del plan o planes recuperados, de tal forma que plantea un problema de satisfacción de restricciones entre el nuevo plan y el plan recuperado que deben satisfacer las variables del nuevo plan.
- ❖ Utilización de heurísticas, empleado cuando se debe cambiar la estructura del plan o planes recuperados. Se suele utilizar junto con otros métodos de adaptación.
- ❖ Basada en casos, consistente en la descomposición de la tarea de adaptación en tareas más pequeñas. Para cada una de estas subtarefas se ejecuta un nuevo ciclo CBP que permita dar solución a la sub-meta correspondiente. Se puede ver como una planificación jerárquica que permite reducir el nivel de complejidad.
- ❖ Planificación generativa, también es un método jerárquico que se utiliza junto con otros métodos de adaptación.
- ❖ Mezcla, en la que se divide el problema en problemas más pequeños que son resueltos de forma independiente para realizar una posterior integración.
- ❖ Derivacional, en la que se siguen los mismos pasos que se siguieron para generar el plan recuperado.
- ❖ Híbrida, consistente en utilizar de forma combinada algunos de los métodos anteriormente propuestos.

---

## 4.2 AGENTE CBR- BDI

---

Como ya se ha comentado, los agentes vienen caracterizados a través de sus capacidades, tales como autonomía, reactividad, pro-actividad, razonamiento, aprendizaje y movilidad entre otras. Las capacidades se pueden modelar de distintas formas y con diferentes metodologías [Wooldridge y Jennings, 1995]. Una de las posibilidades es la de utilizar sistemas de razonamiento basado en casos (CBR).

En los trabajos [Corchado y Laza, 2003][Corchado et al., 2005][Corchado et al. 2008][Bajo et al. 2008] se presenta una formulación que permite relacionar cadenas de creencias con planes de acción y deseos con funciones solución y eficacia de la solución. Esta formulación permite relacionar los conceptos que definen la estructura interna y el comportamiento de un agente BDI con los

conceptos manejados en sistemas de razonamiento basado en casos. Para ello establece un formalismo analítico en el que se proporciona una notación y una relación entre componentes aptitudinales que caracterizan a un agente BDI.

Se denomina  $\theta$  al conjunto que describe el entorno del agente BDI.

Se llama  $T(\theta)$  al conjunto de atributos  $\{\tau_1, \tau_2, \dots, \tau_n\}$  en que el mundo es expresado.

**Definición:** Una creencia  $b$  sobre  $\theta$  es una  $m$ -tupla de atributos de  $T(\theta)$ .  
 $b = \{\tau_1, \tau_2, \dots, \tau_m\}$  donde  $m \leq n$ .

**Definición:** Se llama conjunto de creencias sobre  $\theta$  y se denota por  $\zeta(\theta)$  a  
 $\zeta(\theta) = \{b = \{\tau_1, \tau_2, \dots, \tau_j\} \text{ donde } j = 1, 2, \dots, m \leq n\}$ .

**Definición:** Se introduce el operador “ $\Lambda$  de accesibilidad” entre  $m$  creencias  $b = \{b_1, b_2, \dots, b_m\}$ , y se denota  $\Lambda[b_1, b_2, \dots, b_m] = b_1 \wedge b_2 \wedge \dots \wedge b_m$ , al operador que crea nuevas estructuras uniendo creencias compatibles entre sí.  $\wedge$  es el operador lógico and sobre creencias, que no son otra cosa que expresiones lógicas.

**Definición:** Se dice que el operador actuando sobre creencias es nulo si alguna de ellas no es accesible y se denota como  $\Lambda[b_1, b_2, \dots, b_m] = 0$ .

**Definición:** Una intención  $i$  sobre  $\theta$  es una  $s$ -tupla de creencias compatibles entre sí,  $\{i = (b_1, b_2, \dots, b_m), \text{ donde } \forall i, j \Lambda[b_i, b_j] \neq 0, s \in N\}$ .

**Definición:** Se llama conjunto de intenciones sobre  $\theta$  y se denota  $I(\theta)$  a  
 $I(\theta) = \{(b_1, b_2, \dots, b_k), \text{ donde } k \in N\}$ .

Una vez planteadas estas definiciones, se supone que a partir del análisis de un problema, se determinan un conjunto de parámetros que constituyen una base de atributos sobre la que estructurarlo. Se construye el conjunto  $\varkappa$ , que toma los elementos de  $I(\theta)$  caracterizándolos a partir de atributos de esta base. El conjunto necesario y suficiente de variables se obtiene empíricamente, ya que depende del problema propuesto y depende también en qué términos se quiera resolver.

**Definición:** Se denominan “variables canónicas” de un conjunto  $I(\theta)$  a cualquier conjunto de parámetros linealmente independientes  $\varkappa = (A_1, A_2, \dots, A_v)$  que caractericen los elementos  $i \in I(\theta)$ . Aunque se las denomine variables canónicas, no son únicas, sino que son propias de  $I(\theta)$  para cada problema  $P$  dado.

**Definición:** Se denomina deseo  $d$  sobre  $\theta$  a una función entre:

$$d: I(\Theta) \rightarrow \Omega(\aleph)$$

$$i = (e_1 \wedge e_2 \wedge \dots \wedge e_r) \rightarrow F(A_1, A_2, \dots, A_v)$$

donde  $\Omega(\aleph)$  es el conjunto de funciones sobre  $\aleph$ . Esta expresión no dice otra cosa sino que las actitudes intencionales llamadas deseos pueden ser alcanzadas a través de un plan (que en el esquema utilizado se corresponde con la construcción de una intención adecuada). Para ello se utiliza toda la información que nos proporcionen las creencias, pero evaluadas en términos de los parámetros en los que se especifican los objetivos deseados.

En cuanto a un sistema CBR [Corchado y Laza, 2003] se puede definir una formulación para los componentes del sistema como sigue:

**Definición:** Un conjunto de bases de casos ( $\beta$ ). Una base de casos  $B \in \beta$  es un conjunto finito de casos que se encuentra indexado. Una base de casos se define como un tupla  $(\{c_1, c_2, \dots, c_n\}, \iota)$ .  $\{c_1, c_2, \dots, c_n\}$  son los casos que componen la base de casos y  $\iota$  es el conjunto finito de características que permite indexar los casos.

**Definición:** Un caso ( $c$ ) representa una experiencia pasada. Un caso se representa mediante una secuencia de estados del entorno:  $c = \{\text{estado\_inicial}, \{\text{acción } x[\text{estado\_intermedio}]\}^+, \text{estado\_final}\}$ . Cada estado se representa a través de un conjunto de atributos que define el entorno en el que se encuentra situado el sistema CBR. Los estados se dividen en tres grupos:

- Conjunto de estados iniciales ( $\text{estado\_ini}$ ), que representan la descripción del problema que debe ser resuelto
- Conjunto de estados intermedios ( $\text{estado\_inter}$ ), que describen los distintos estados por los que pasa el entorno antes de alcanzar el estado final
- Conjunto de estados finales ( $\text{estado\_final}$ ), que representa la descripción del entorno una vez que se ha alcanzado los objetivos iniciales.

Además de estados un caso contiene acciones, que representan el conjunto de acciones aplicadas a cada uno de los estados. Se definen mediante un nombre y un conjunto de argumentos.

**Definición:** Un conjunto finito de atributos ( $k$ ) son un conjunto de propiedades que permiten describir un estado.

**Definición:** Un conjunto de índices ( $I$ ) es un conjunto de características de  $\iota$ , con  $\iota$  incluido en  $k$ .

**Definición:** Un conjunto de funciones de similitud ( $A$ ) permiten determinar el grado de similitud que existe entre un problema a resolver y un caso.

El ciclo del sistema CBR queda definido a través de la Tabla 2. En ella, es posible apreciar como en la etapa de recuperación se obtiene de la memoria de casos B aquellos casos  $c_1, c_2, \dots, c_k$ , con una descripción de problema más similar al problema actual  $st_n$  utilizando para ello una métrica A. Se divide en dos fases, la de indexación y la de selección. En la etapa de reutilización se obtiene una primera solución  $(\{act_{ni}, \{st_{inter\_ni}\}^+, st_{final\_n})$  a partir de los casos recuperados y de la descripción del problema. En la etapa de revisión se evalúa la validez de la solución propuesta para, finalmente, en la etapa de aprendizaje aprender de la nueva experiencia.

**Tabla 2. Ciclo de un sistema CBR.**

Recuperación	$(c_1, c_2, \dots, c_k, A) \leftarrow \text{Retrieve}(st_n, B)$ with $c_k = \{st_k, \{act_{ki}, \{st_{inter\_ki}\}^+, st_{fin\_k}\}, k > 0 \text{ and } i > 0$
Reutilización	$(st_n, \{act_{ni}, \{st_{inter\_ni}\}^+, st_{final\_n}) \leftarrow \text{Reuse}(st_n, (c_1, c_2, \dots, c_k), A)$
Revisión	$(st_n, \{act_{ni}, \{st_{inter\_ni}\}^+, st_{final\_n}) \leftarrow \text{Revision}(st_{fin\_n})$
Aprendizaje	$(st_n, \{act_{ni}, \{st_{inter\_ni}\}^+, st_{final\_n}, B) \leftarrow \text{Learning}(c_n, B)$

Los deseos previos, junto con creencias e intenciones se almacenan en forma de casos como se muestra a continuación:

<b>Caso: &lt;Problema, Solución, Resultado&gt;</b>	<b>Agente BDI</b>
Problema: estado_inicial	Creencia: estado
Solución: secuencia de <acción, [estado_intermedio]>	Intención: secuencia de <acción>
Resultado: estado_final	Deseo: conjunto de <estado_final>

Una creencia es un estado que puede ser inicial (representa el problema que debe ser resuelto), intermedio (representa un estado intermedio del problema por el que se pasa antes de alcanzar el estado final) o final (representa el resultado obtenido después de partir del estado inicial y realizar un conjunto de acciones). Cada creencia posee una serie de atributos que la describen. Un deseo es un conjunto de estados finales. Dependiendo del problema un deseo podrá estar formado por uno o más estados finales que el agente quiere alcanzar. Una intención es una secuencia ordenada de acciones. Las acciones son operaciones que se realizan sobre un determinado estado. Cada acción se define a través de un nombre y de un conjunto de argumentos.

Una especialización del agente CBR-BDI es el agente CBP-BDI [Corchado et al., 2008]. Su estructura interna es similar, pero difiere del agente CBR-BDI en que la solución que lleva a cabo el agente no es un modelo, si no un plan, es decir, una secuencia de acciones. Un agente CBP-BDI debe incorporar un

mecanismo de planificación que genere planes como soluciones a los problemas planteados. En un agente CBP-BDI es posible incorporar diferentes mecanismos de planificación. En [Corchado et al., 2008] se presenta un mecanismo de replanificación para los agentes CBP-BDI cuya principal característica es la de ser capaz de replanificar en tiempo de ejecución. Un agente CBP-BDI calcula el plan o intención que es más fácilmente replanificable (MRPI). Esto es, aquel plan que puede ser más fácilmente sustituible por otro plan en caso de que se vea interrumpido (por ejemplo si un usuario cambia sus preferencias durante la ejecución del plan).

En una sociedad de agentes abierta, disponer de planes alternativos es muy importante para mantener la eficiencia del sistema. La funcionalidad principal de una sociedad es permitir a sus miembros coexistir en un entorno compartido y llevar a cabo sus respectivos objetivos cooperando o no con el resto de miembros. Esto constituye un problema abierto, en el que los sistemas clásicos de planificación no son apropiados. Se requiere un modelo que pueda adaptarse a estas circunstancias. En el capítulo 5 se mostrará el modelo propuesto en esta tesis, el cual hará uso de las capacidades de los agentes CBP-BDI.

---

### 4.3 AGENTES VIRTUALES INTELIGENTES

---

Para la simulación del comportamiento de los agentes se utilizará un entorno de AVIs [Aylett y Luck, 2000] [Lozano y Calderón, 2006] (agentes virtuales inteligentes) en donde agentes inteligentes representados en modelos tridimensionales forman parte de un entorno virtual simulado [West y Hubbard, 1998].

Un agente lo podíamos ver como una entidad software que, basándose en su propio conocimiento del entorno, realiza una serie de operaciones destinadas a realizar un objetivo, ya sea por iniciativa propia o porque dada una situación se requiera. Los agentes inteligentes pueden considerarse como entidades individuales dentro de un programa, en nuestro caso un entorno virtual, que tienen control sobre su vida y movimientos dentro del mismo. Un *Agente Virtual Inteligente (AVI)* posee características propias de los *Entornos Virtuales* [Aylett & Luck, 2000] y de los Agentes Inteligentes:

- ❖ Habitan dentro de un entorno de ejecución 3D simulado.
- ❖ Poseen una representación gráfica 3D dentro del mundo que habitan, y son capaces de percibir, adaptarse y reaccionar a su entorno.
- ❖ Son capaces de expresar sus comportamientos de manera gráfica como lo haría un ser vivo.

- ❖ Aunque solo existen y funcionan en un entorno específico, son conscientes de los cambios que se producen a su alrededor y son capaces de responder a ellos de manera autónoma.

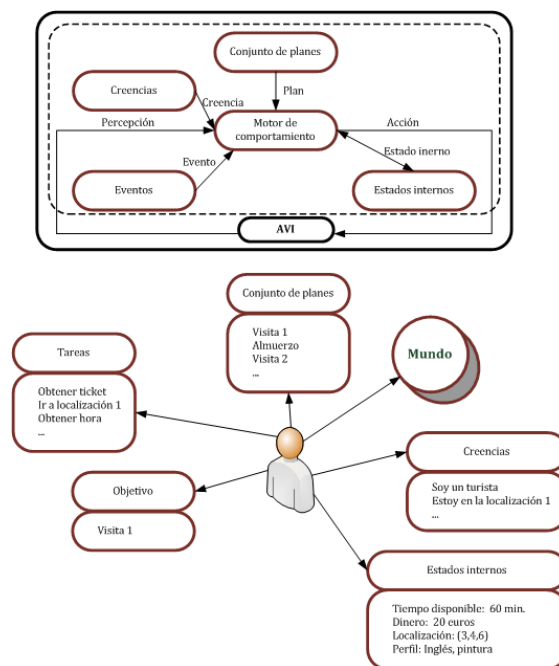


Figura 9. Agente Virtual Inteligente

El comportamiento (animación) asociado a los estos agentes tiene por objetivo hacerlo lo más realista posible. Existe un conjunto de técnicas de IA que pueden ayudar a alcanzar dicho propósito [Caicedo y Thalmann, 2000] [Becheiraz y Thalmann, 1998]. Los comportamientos complejos son de especial atención en este tipo de ambientes virtuales inteligentes, dichos comportamientos pueden ser divididos en:

- ❖ Motivados por emociones, en donde las emociones sentidas por un actor son causadas por la percepción de objetos, acciones y/o eventos; y están caracterizadas por condiciones para cada emoción junto variables que miden su intensidad.
- ❖ Motivados por creencias o razonamientos, en donde las conductas de los agentes son orientadas a objetivos, presentando un proceso de toma de decisiones compuesto de un planificador de objetivos, manejo dinámico de creencias, objetivos que evolucionan, estados internos, y niveles de confianza.

Los trabajos de [Caicedo y Thalmann, 2000] intentan conjugar técnicas de IA para formar lo que se conoce como *Marco de Simulación del Comportamiento*.

Independientemente del diseño del mundo virtual en el que habitan, los agentes virtuales poseen módulos de comportamiento que les permiten reconocer el entorno a su alrededor y ofrecer respuestas acorde a los estímulos, tanto externos como internos, que reciben. Posteriormente sus trabajos evolucionaron hacia un *Marco de Simulación del Comportamiento* más elaborado para construir humanos virtuales con conductas orientadas a objetivos, presentando un proceso de toma de decisiones compuesto de un planificador de objetivos, manejo dinámico de creencias, objetivos que evolucionan, estados internos, y niveles de confianza [Conde y Thalmann, 2006a][Conde y Thalmann, 2006b]. Sus trabajos estaban más relacionados con el concepto de AVA (Agentes Virtuales Autónomos), en los que los agentes están constituidos por un conjunto de técnicas que, interrelacionadas adecuadamente, les permiten un comportamiento lo más parecido y realista comportamiento humano. Del estudio de las reacciones de los AVA, se están sacando conclusiones de las reacciones del comportamiento humano, ya que posibilitan el estudio de patrones de comportamiento de forma visual.

En los últimos años, se han integrado distintas arquitecturas de agente para el soporte de estos requerimientos comportamentales. Así, [Terzopoulos , 1999] ha desarrollado uno de los ejemplos más conocidos de vida artificial: los peces artificiales, agentes autónomos con características de un pez real, con su propio movimiento, percepción, comportamiento y aprendizaje. Otro ejemplo es el humano virtual de Jack [Badler y Palmer, 1999], un de los primeros personajes usado principalmente para el estudio de modelos biomecánicos, aunque integra un sistema de planificación completo para la animación de su comportamiento.

En este trabajo de tesis, las conductas de los agentes son orientadas a objetivos, presentando un proceso de toma de decisiones del planificador de la organización. El comportamiento de la organización, utilizando el del modelo del capítulo 5 es simulado en un entorno dinámico 3D en el caso de estudio del capítulo 6. Los agentes de la organización definirán su comportamiento en la plataforma THOMAS, para posteriormente ser representados en un entorno 3D. En este tipo de entornos es posible ampliar e integrar una mayor interacción con entes virtuales. Para el módulo de simulación se ha utilizado REPAST (REcursive Porous Agent Simulation Toolkit) [North et al., 2007][REPAST, 2009], un conjunto de librerías Java que permiten a los desarrolladores crear entornos de simulación, crear agentes en redes sociales, recopilar datos de las simulaciones automáticamente, y construir interfaces de usuarios fácilmente.





## PARTE III. PROPUESTA



---

## 5 MODELO DE ORGANIZACIÓN

---

En este capítulo se presenta el modelo para organizaciones virtuales propuesto. Surge de la idea de que es necesario un modelo de coordinación que facilite la auto-adaptación de una sociedad de agentes. Tendremos un SMA cooperativo, en el cual cada agente es capaz de establecer planes dinámicos para alcanzar sus objetivos; además de un mecanismo global que permite realizar una asignación óptima de actividades a los agentes para que puedan trabajar de forma coordinada. Este mecanismo tendrá en cuenta el objetivo global de la sociedad, así como sus normas y roles.

Los agentes tienen capacidades limitadas que dificultan la posibilidad de tener un conocimiento completo y preciso del entorno y los efectos de las acciones de los agentes pueden variar por distintos motivos (cambios en la estructura interna de los agentes, en las normas, etc.). El comportamiento de los agentes estará determinado por las metas individuales que desee alcanzar, pero teniendo en cuenta el resto de metas de los agentes compañeros y los cambios en el entorno.

El capítulo está organizado de la siguiente manera. La sección 5.1 proporciona una introducción en la que se muestra la motivación que dio lugar al desarrollo del modelo en una organización virtual, en particular la idea de que una organización puede adaptar sus acciones frente a cambios para conseguir sus objetivos. En la sección 5.2 se hace un resumen de los trabajos relacionados con la adaptación de organizaciones, mostrando las ventajas y las desventajas de cada uno de ellos. La formalización del modelo es presentada en la sección 5.3 y finalmente, la sección 5.4 describe líneas de trabajo futuro y presenta algunas de las conclusiones obtenidas.

Este capítulo extiende los trabajos sobre SMA presentados en [Corchado et al., 2006][Bajo et al., 2007][Corchado et al. 2008][Rodríguez et al., 2009] sobre la base de las ideas presentadas en los proyectos THOMAS (TIN2006-14630-C03-03.) y OVAMAH (TIN 2009-13839-C03-03).

## 5.1 MOTIVACIÓN

---

Dentro del estudio de las organizaciones a nivel de agentes, encontramos un conjunto de requisitos [Rodríguez et al., 2009] que incitan a desarrollar nuevos modelos sociales en los que la utilización de sistemas abiertos [Zambonelli et al., 2003] y adaptativos [Di Marzo et al., 2004] sea posible. Hemos visto cómo los sistemas abiertos se caracterizan por la heterogeneidad de sus participantes, la confianza limitada, objetivos individuales en conflicto y una gran probabilidad de disconformidad con las especificaciones [Giret et al., 2005]. La adaptación al entorno es un reto conocido en todas las investigaciones más recientes [Zambonelli et al., 2004][Reitbauer et al., 2004][Weyns et al., 2004] [Capera et al., 2003][Razavi et al., 2005]. Se pretenden sistemas software capaces de responder y actuar por sí mismos ante cambios en su entorno. Para ello se hace necesario disponer de teorías, modelos, mecanismos, métodos y herramientas que permitan desarrollar sistemas con capacidad de reorganización y que puedan adaptarse de esta forma a futuros cambios en su entorno.

Por otro lado, veíamos cómo cada organización necesita de un soporte de coordinación que determine explícitamente cómo deben organizarse los agentes y llevar a cabo las acciones y tareas asignadas dentro de la misma. Esta coordinación está relacionada con la planificación de acciones para la resolución de tareas, porque estos planes permiten: (i) conocer el comportamiento de otros agentes del sistema; (ii) intercambiar resultados intermedios que lleven al progreso en la solución a la tarea global; y (iii), evitar acciones redundantes.

Durante los capítulos precedentes, se ha hecho énfasis en la importancia de comprender las nociones tanto organizativas, como de planificación y adaptación, ya que el modelo que se presenta a continuación contiene, como elemento fundamental, la utilización una planificación de agentes inmersos en entornos altamente cambiantes que proporciona la posibilidad de replanificar en tiempo de ejecución.

La noción de **organización** viene dada por la utilización del concepto social para el desarrollo de SMA, utilizando para ello la plataforma THOMAS. La noción de **planificación** empleada en este documento, hace referencia a su acepción más abstracta, abordando la construcción de un modelo propio que permita la generación de un plan. Se requiere de un modelo que pueda llevar a cabo una planificación adaptativa dentro de una organización de agentes y pueda aplicarse a entornos reales. Como vimos en el apartado 2.1.5, existen diferentes paradigmas de planificación, pero el más adecuado para llevar a cabo este objetivo es la planificación basada en casos ya que nos proporciona la flexibilidad necesaria para llevar a cabo una replanificación de las tareas, acciones y objetivos de la organización. Y por último, la noción de **adaptación**

---

viene dada por la utilización de mecanismos basados en interacciones directas [Zambonelli et al. 2004]: interacciones y cálculos locales para reorganizar los agentes para obtener un estado global coherente del sistema.

Siguiendo estas ideas surge la motivación para el diseño de un modelo integrador en el que una organización de agentes pueda ejecutar acciones basándose en un modelo de planificación social. La innovación en él consistirá en dotar a una organización de capacidades auto-adaptativas en tiempo de ejecución para entornos altamente dinámicos. De esta forma, el comportamiento de un agente estará determinado por las metas que desee alcanzar, pero teniendo en cuenta el resto de metas de los agentes de la organización y los cambios en el entorno.

---

## 5.2 TRABAJOS RELACIONADOS

---

Las organizaciones de agentes dependen del modo de coordinación, del tipo de comunicación, entre agentes, así como del tipo de agentes que conforman el grupo. En general se pueden distinguir diferentes enfoques organizacionales: MOISE [Boissier 2007], OperA [Dignum 2004], INGENIAS [Pavon 2003], the Gaia Methodology [Zambonelli et al., 2003], [Omicini et al., 04]. Pero mientras estos trabajos proporcionan mecanismos para la coordinación entre participantes, existe mucho menos trabajo centrado en la adaptación en tiempo de ejecución de las estructuras organizacionales o normas definidas en el diseño. Por ejemplo [Hubner, 2004] propone un modelo para controlar la adaptación creando nuevas normas. [Gasser y Ishida 1991] proponen un modelo distribuido para reorganizar su arquitectura. [Artikis et al., 2009] hace que los agentes sigan un protocolo para adaptar sus normas. Todos estos trabajos están centrados en la estructura y/o normas, basando su adaptación en la coordinación entre participantes.

Existen otros enfoques como los basados en normas sociales (en el rol del grupo social y efecto en la aparición y apoyo de normas sociales) [Hales 2002][Lakkaraju y Gasser, 2008]. Otra posibilidad es el desarrollo de SMA con un enfoque basado en el concepto de organización/institución. Las interacciones humanas están reguladas por instituciones [North, 1990] que representan las reglas del juego en una sociedad y definen lo que los individuos pueden y no pueden hacer y bajo qué condiciones. Esta perspectiva defiende la adopción de una estrategia mimética con el objeto de afrontar la complejidad del desarrollo de SMA abiertos bajo un concepto organizacional. Una institución electrónica [Noriega et al., 1999][Rodríguez-Aguilar, 2003][Esteva, 2003][García et al., 2009] debe ser considerada como un middleware social entre los agentes externos participantes y la capa de comunicación escogida encargada de aceptar o rechazar las acciones de los agentes. También existen trabajos que direccionan este tema a la definición de "Automatic Electronic

Institutions”, una extensión de las instituciones electrónicas dotadas con capacidades de auto-adaptación [Bou et al., 2007a][Bou et al., 2007b][Bou et al., 2009]. La principal diferencia con las otras propuestas es que la adaptación es llevada a cabo por la institución, en lugar de por los agentes.

Otros puntos de vista [Shoham, 1992][Shoham y Tennenholtz, 1997][Kittock, 1993] [Walker, 1995] se centran en mecanismos de regulación de comportamientos. Aquí el factor clave es la "imitación", es decir, las convenciones sociales ocurren a través de propagación y contagio [Conte y Paolucci, 2001][Burt, 1987].

Por último existen enfoques en los que los agentes utilizan la información obtenida durante sus interacciones [Axelrod 1986][Castelfranchi y Conte, 1995][Sen y Airiau, 2007]. En estos trabajos las normas sociales (como tipos especiales de convenciones sociales) son usadas por los agentes para auto-regular su comportamiento según la información social: historia previa, reputación, etc. La investigación se centra en mecanismos de grupos sociales basados en la información social recopilada durante las interacciones [Villatoro y Sabater-Mir, 2008a][ Villatoro y Sabater-Mir,2008b][Villatoro y Sabater-Mir,2008c]. Investigan los efectos de los algoritmos de transmisión de información para reconocer y formar grupos sociales.

En ninguno de los enfoques estudiados se lleva a cabo una coordinación de tareas de los agentes miembros de la organización que se combinen para resolver un problema común, y que además, tengan en cuenta que la planificación de las tareas debe adaptarse a los cambios del entorno. El modelo social utilizado en la arquitectura elegida, THOMAS [Carrascosa et al., 2009] [Giret et al., 2009], está centrado en la definición de la estructura y las normas. La adaptación en este tipo de modelos se basa en la coordinación entre los participantes de la organización. Vamos a ver a continuación el modelo propuesto con el que se pretende llevar a cabo una planificación adaptativa dentro de la organización de agentes.

---

### *5.3 MODELO DE PLANIFICACIÓN ADAPTATIVA*

---

La actividad central de este trabajo consiste en la formalización de un modelo de planificación capaz de integrarse en un agente deliberativo, de tal forma que dote a los agentes de una organización virtual de un mayor grado de autonomía, de adaptación al entorno y de capacidad de reorganización. La arquitectura descrita en la sección 3.1.2.1, está preparada para el desarrollo de organizaciones, evoluciona al describir el modelo, permitiendo que los sistemas construidos tengan capacidad de adaptación y coordinación.

---

En la sección 5.3.1 se presenta la arquitectura de agente deliberativo CBP-BDI propuesto para la construcción del modelo. En la sección 5.3.2 se detalla el mecanismo de planificación global propuesto así como el modelo de planificación utilizado por el agente CBP-BDI para alcanzar sus objetivos individuales.

---

### 5.3.1 Agentes CBP- BDI

---

Si bien existen distintas arquitecturas para agentes que intentan dar una respuesta satisfactoria a los objetivos planteados, percibíamos cómo un modelo particularmente interesante es el basado en las ideas de razonamiento deliberativo [Bratman et al., 1988] [Corchado 2004] que ha dado origen a las arquitecturas BDI. En un agente BDI, se pone especial énfasis al razonamiento dirigido a la acción, es decir, a determinar qué debe hacer el agente [Wooldridge 2002]. Este proceso involucra dos actividades fundamentales: a) determinar qué metas se desean alcanzar (deliberación) y b) decidir de qué manera alcanzar estas metas (planificación). Ambos procesos deben realizarse considerando las limitaciones de recursos de los agentes, es decir, que los procesos de razonamiento deben tener en cuenta las restricciones impuestas en el sistema para seleccionar las acciones en forma efectiva.

Las arquitecturas BDI exhiben propiedades interesantes que pueden permitir abordar esquemas de coordinación de alto nivel con los agentes disponibles en un momento determinado y plantear las extensiones necesarias a este modelo, de manera tal que los agentes tengan un comportamiento social adecuado. El modelo presentado incluye un tipo de agentes con capacidad de planificación desde el punto de vista de una organización de agentes. El tipo de agente en concreto es un agente deliberativo CBR-BDI [Corchado y Laza, 2003]. La característica principal de estos agentes, como hemos visto el capítulo 4, es que integran un mecanismo de razonamiento CBR [Admond y Plaza, 1994]. Un sistema CBR ejecuta un ciclo de razonamiento formado por cuatro etapas secuenciales: recuperación, adaptación, revisión y aprendizaje. Este ciclo está integrado dentro de las actividades del agente BDI [Corchado y Laza, 2003] identificando las etapas como tareas o roles que el agente debe ser capaz de llevar a cabo. De esta forma se suple una de las principales carencias del modelo BDI que es el manejo de la memoria y de las experiencias pasadas. En [Bajo et al., 2008], [Carrascosa et al., 2008], [Corchado et al., 2008], [Glez-Bedia, 2004] se presenta el método de incorporación del motor CBR al modelo BDI. La idea principal de estos trabajos es la de utilizar los mecanismos que proporciona el modelo deliberativo BDI, como son creencias (*Beliefs*), deseos (*Desires*) e intenciones (*Intentions*), para poder obtener una representación de caso y para poder ejecutar el ciclo de razonamiento CBR.

Para poder integrar el sistema de razonamiento CBR dentro de la estructura de un agente deliberativo BDI, se plantea una formulación que permite relacionar

el concepto de caso con los conceptos fundamentales BDI [Corchado et al., 2008], [Glez-Bedia, 2004].

La formalización llevada a cabo para la integración de las dos ideas es la siguiente.

El entorno  $M$  y los cambios que se producen en dicho entorno se representan desde el punto de vista del agente. Así, el “mundo” del agente puede definirse como un conjunto de variables que influyen sobre los problemas que intenta resolver el agente.

$$M = \{\tau_1, \tau_2, \dots, \tau_s\} \text{ with } s < \infty \quad (5.1)$$

Las creencias son objetos formados por algunos (o todos) de los atributos del mundo, y toman valores concretos.

$$B = \{b_i / b_i = \{\tau_1^i, \tau_2^i, \dots, \tau_n^i\}, n \leq s \quad \forall i \in N\}_{i \in N} \subseteq M \quad (5.2)$$

Un estado del mundo  $e_j \in E$  se representa como un conjunto de creencias que son ciertas en cierto instante de tiempo  $t$ . Si  $E = \{e_j\}_{j \in N}$  es el conjunto de estados del mundo en un instante concreto de tiempo  $t$ , entonces es posible obtener la ecuación :

$$e_j^t = \{b_1^{jt}, b_2^{jt}, \dots, b_r^{jt}\}_{r \in N} \subseteq B \quad \forall j, t \quad (5.3)$$

Los deseos u objetivos se imponen al comienzo de un problema, pero pueden cambiar a lo largo del tiempo. Se trata de aplicaciones que relacionan un estado del mundo actual y otro estado que se trata de alcanzar.

$$d : \underset{e_0}{E} \rightarrow \underset{e^*}{E} \quad (5.4)$$

Las intenciones son aplicaciones que indican la forma en la que el agente utiliza sus conocimientos para alcanzar sus objetivos. En el modelo CBR-BDI las intenciones deben determinar si existe compatibilidad entre los conocimientos que posee el agente y los requisitos necesarios para ser capaz de alcanzar sus deseos. De esta forma un deseo será alcanzable si se cumple la expresión presentadas en la ecuación:

$$i : \underset{(b_1, b_2, \dots, b_n, e_0)}{B \times B \times \dots \times B \times E} \xrightarrow{n)} \underset{e^*}{E} \quad (5.5)$$

En el modelo CBR-BDI, las intenciones garantizan que se dispone de suficiente conocimiento en la base de creencias para ser capaz de alcanzar un deseo, proporcionando una solución.



Esta formulación permite relacionar cadenas de creencias con planes de acción y deseos con funciones solución y eficacia de la solución. Para integrar un mecanismo de razonamiento basado en casos en un agente BDI, el planteamiento que se sigue presenta un agente  $A = \{\{b_{ij}\}\{d_{ij}\}\}$  que se enfrenta a un problema  $P = (S_i, S_f)$  que debe solucionar para alcanzar sus objetivos. El agente necesita una intención  $\{i_j\}$ , esto es el uso adecuado de información combinada que posee en forma de creencias, para identificar la intención que constituye una solución  $S(P)$ . El agente consulta una memoria de casos tratando de encontrar una solución similar que ya hubiese sido utilizada en el pasado para resolver un problema de características similares a las de  $S(P)$ . El agente obtendrá la solución que considere más adecuada  $S(P)$ , y a partir de ella construye una intención o plan capaz de poner en marcha dicha solución. Es necesario tener en cuenta que tanto el problema como la descripción del entorno almacenados en un caso van a poder ser similares al actual, pero que al admitir dinamismo, nada nos dice que la evolución del entorno en el pasado que hizo de  $P$  una buena solución para  $S(P)$  vaya a serlo en este caso.

Así pues, se dispone de un agente BDI con una serie de creencias que se encuentran almacenadas, junto con una serie de planes ejecutados en el pasado, en un motor de casos CBR. Formalmente se puede describir este sistema como:

1. Dada una base de casos o memoria de casos

$$C = (P, S(P), Eff) \quad (5.6)$$

Donde los problemas vienen definidos como  $P = (S_i, \{\theta_j\}, S_f)$ , siendo  $S_i$  el estado inicial del problema y  $S_f$  el estado final o estados objetivo que se desea alcanzar.  $\{\theta_j\}$  con  $j = 1, 2, \dots, m$  es el conjunto de operadores a partir de los que se construye la solución.

2. Las soluciones para los problemas estudiados, para las que se utiliza una representación matricial.  $S(P) = \{S_{k_i}, \theta_{h_i}\}$ , donde  $k = 1, 2, \dots, n+1$  y  $h = 1, 2, \dots, n$ , con lo que  $S_1 = S_i$  y  $S_{n+1} = S_f$ .

$$S(P) = \{S_1, \theta_1, S_2, \theta_2, \dots, \theta_n, S_{n+1}\} \quad (5.7)$$

que expresado en forma matricial:

$$S(P) = \left\{ \begin{matrix} O(S_i) \\ R(S_i) \end{matrix}, \begin{matrix} O(S_f) \\ R(S_f) \end{matrix} \right\} \quad (5.8)$$

donde los estados  $O = O(O_1, O_2, \dots, O_n)$  es el vector de objetivos y  $R = R(R_1, R_2, \dots, R_n)$  es el vector de recursos disponibles. Se utiliza esta

notación teniendo en cuenta que los estados  $S_k$  y los operadores  $\Theta_j$  se definen como:

$$S_k = \begin{pmatrix} \{O_r\}_{r=1,2,\dots,p} \\ \{R_s\}_{s=1,2,\dots,q} \end{pmatrix} \quad (5.9)$$

$$\Theta_j : S_k = \begin{pmatrix} \{O_r\} \\ \{R_s\} \end{pmatrix} \rightarrow \Theta_j(S_k) = \begin{pmatrix} \{O'_r\} \\ \{R'_s\} \end{pmatrix}$$

Siendo  $\{O_r\}_{r=1,2,\dots,p}$  las variables con las que se representan los estados del problema.

3. La eficiencia o eficacia de una solución viene dada a través de restricciones sobre las relaciones que se establecen entre los objetivos a alcanzar y los recursos disponibles para alcanzar dichos objetivos.

$$\left. \begin{aligned} f(O_1, O_2, \dots, O_n) &= 0 \\ g(R_1, R_2, \dots, R_m) &= 0 \\ h(O_1, O_2, \dots, O_n, R_1, R_2, \dots, R_m) &= 0 \end{aligned} \right\} \quad (5.10)$$

Se tiene entonces una relación que permite integrar los sistemas CBR en arquitecturas BDI. Dicha relación asocia creencias, deseos e intenciones con casos y su ventaja es que cualquier problema puede ser fácilmente conceptualizado en términos de agentes e implementado posteriormente en forma de CBR [Corchado y Laza, 2003].

Una especialización del agente CBR-BDI es el **agente CBP-BDI** [Corchado et al., 2008]. Su estructura interna es similar, pero difiere del agente CBR-BDI en que la solución que lleva a cabo el agente no es un modelo, si no un plan, es decir, una secuencia de acciones. Un agente CBP-BDI debe incorporar un mecanismo de planificación que genere planes como soluciones a los problemas planteados. En un agente CBP-BDI es posible incorporar diferentes mecanismos de planificación [Bajo et al., 2007], [Bajo et al., 2008], [Corchado et al., 2007], [Corchado et al., 2008]. En [Corchado et al., 2008] se presenta un mecanismo de replanificación para los agentes CBP-BDI cuya principal característica es la de ser capaz de replanificar en tiempo de ejecución. Un agente CBP-BDI calcula el plan o intención que es más fácilmente replanificable (MRPI). Esto es, aquel plan que puede ser más fácilmente sustituible por otro plan en caso de que se vea interrumpido (por ejemplo si un usuario cambia sus preferencias durante la ejecución del plan).

---

En un entorno dinámico, como ocurre con las sociedades de agentes, disponer de planes alternativos es muy importante para mantener la eficiencia del sistema. Como hemos visto, cada vez es más frecuente modelar SMA no solamente desde el punto de vista individual del agente y sus capacidades de comunicación, si no que es necesario una ingeniería organizacional. La funcionalidad principal de una sociedad es permitir a sus miembros coexistir en un entorno compartido y llevar a cabo sus respectivos objetivos cooperando o no con el resto de miembros. La coordinación de tareas que se combinan para resolver un problema común, requiere de una planificación centralizada, o bien de una planificación distribuida llevada a cabo por los propios agentes del sistema. Esto constituye un problema abierto, en el que los sistemas clásicos de planificación no son apropiados. Se requiere de una planificación que pueda adaptarse a estas circunstancias.

---

### 5.3.2 Modelo de planificación para un agente CBP-BDI en una OV

---

El objetivo principal que se pretende conseguir en este estudio es conseguir un modelo de planificación que permita, dentro de una arquitectura orientada al desarrollo de organizaciones de agentes (THOMAS [Giret et al. 2009][Carrascosa et al. 2009]), considerar los cambios que se producen en el entorno durante la ejecución de un plan. La particularidad principal que hay que considerar en este caso es que los planes serán ejecutados por los agentes miembros de una organización. Los planes deberán tener en cuenta no solamente las restricciones impuestas a un agente individual, si no los compromisos, restricciones y normas de la organización. La planificación deberá definir las acciones que la sociedad de agentes tendrá que ejecutar, y por lo tanto, deberá también tener en cuenta las particularidades de todos sus miembros. Es una planificación adaptativa dentro de una organización de agentes. Para ello se utilizará un tipo de agente CBP-BDI como el descrito anteriormente aplicando el siguiente modelo de planificación adecuada a organizaciones.

Un plan  $p$  dentro de una organización, está definido como  $p = \langle E, O, O', R, R' \rangle$ , donde:  $E$  es el entorno, que representa el tipo de problema que resuelve la organización, caracterizado por un conjunto de estados  $E = \{e_0, e^*\}$  para cada agente, donde  $e_0$  representa el estado inicial del agente cuando comienza un plan, y  $e^*$  es el estado o conjunto de estados que el agente trata de alcanzar.  $O$  representa el conjunto de objetivos del agente de manera individual y  $O'$  es el conjunto de objetivos alcanzados una vez ejecutado el plan.  $R$  es el conjunto de recursos disponibles para ese agente y  $R'$  es el conjunto de recursos que ha consumido el agente durante la ejecución del plan.

Dado un estado inicial de la organización, el término problema de *planificación global* se utiliza para describir la búsqueda de una solución que permita alcanzar un estado final que cumpla con una serie de requisitos de la organización. El problema puede ser representado en un espacio de

planificación, que estará acotado por las restricciones impuestas por los requisitos.

La finalidad es encontrar el plan global que permita a la organización, dado un objetivo común, con una serie de tareas a llevar a cabo y recursos disponibles, encontrar la solución óptima. Para hacer esto, el agente coordinador deberá tener en cuenta los planes óptimos  $p^*(t)$  obtenidos para cada agente individual. No todos los agentes de la organización tienen por qué saber cumplir los objetivos, pero sí llevar a cabo algunas de las tareas que haga que se llegue al objetivo de la organización. Se generarán diversos planes que resolverán de forma genérica el problema y posteriormente serán descompuestos en distintas tareas que serán distribuidas entre los agentes del grupo.

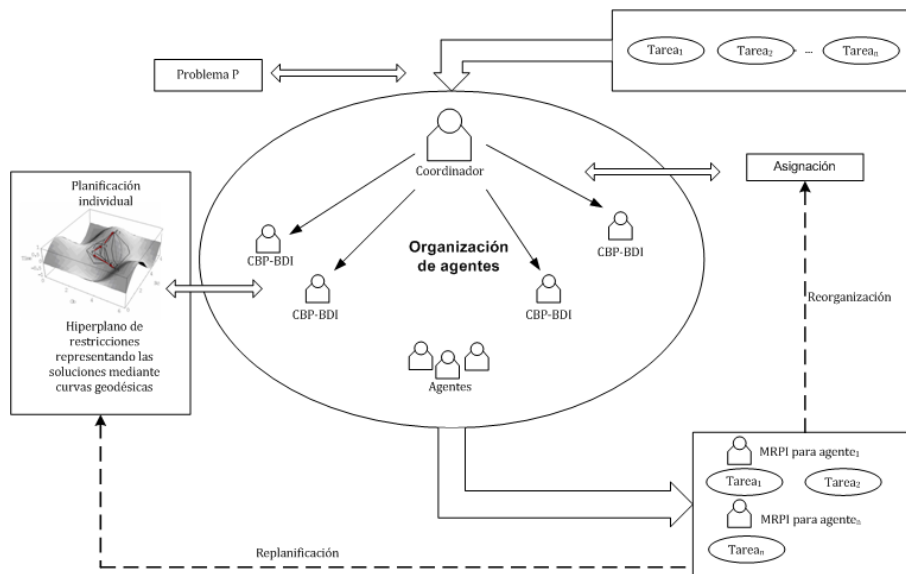


Figura 10. Modelo de planificación

Al iniciar el proceso, se recuperan de la base de datos aquellos agentes que al menos hagan una de las tareas del problema. Por cada tarea que no fuese hecha por ninguno de los agentes recuperados, se incorporaría como mínimo un nuevo agente; el cual tendría asignada la mayor probabilidad de cumplir con éxito dicha tarea. La idea es que se cuente con los agentes necesarios para que no pueda quedar ninguna tarea sin asignación; es decir, que la unión de las tareas que pueden hacer los agentes, sea como mínimo el conjunto de las tareas a llevar a cabo, para cumplir el plan.

Supongamos que el objetivo común de los “m” agentes tiene “n” estados o tareas, con  $m, n \in \mathbb{N}$ .

Cada agente tiene unas características propias en cuanto a tareas que puede hacer, recursos que usa y tiempo que emplea en realizar las tareas; es decir, cada agente tiene su perfil.

Fijado un estado "j", para cada agente "i" con  $i \in \{1, \dots, m\}_{m \in \mathbb{N}}$ , se le puede identificar con una  $z_{i,j}$ -tupla, donde cada coordenada de la tupla se refiere a una característica que lo define.

Se definen las siguientes variables binarias:

$$a_{i,j} = \begin{cases} 1 & \text{si al agente "i" se le asigna la tarea "j"} \\ 0 & \text{en otro caso} \end{cases} \quad (5.11)$$

En todos los problemas de asignación, se define una función objetivo que puede tener como meta tanto minimizar como maximizar el tiempo empleado por los "m" agentes para llevar a cabo el objetivo común. Por ejemplo, minimizar o maximizar el coste del uso de los agentes para cumplir con el objetivo, o maximizar una función de eficiencia, según convenga en cada caso.

Para la asignación de tareas a los agentes, se introduce una nueva función de eficiencia, con ello buscamos visitar el mayor número de puntos con el menor coste posible. En la función de eficiencia definida en los trabajos previos [Corchado et al, 2008], lo que se buscaba era que, una vez asignadas las tareas a los agentes, la ruta a seguir fuera la mejor para cumplir el plan. La nueva función de eficiencia, depende del número de puntos visitados y del coste total que ello suponga.

El coste es otra función que depende a su vez, del tiempo que esté trabajando el agente "i" en la tarea "j", de los recursos empleados y del tipo de agente asignado a cada tarea; lo cual se denota por:  $c_{t_{ij} r_j}^i$ .

Se define la función de eficiencia como:

$$Eficiencia = \frac{N^{\circ} \text{ puntos visitados}}{\sum_{i=1}^m \sum_{j=1}^n c_{t_{ij} r_j}^i \cdot a_{i,j}} \quad (5.12)$$

Supongamos que queremos maximizar esa función de eficiencia:

$$\text{Max} \frac{\text{N}^\circ \text{ puntos visitados}}{\sum_{i=1}^m \sum_{j=1}^n c_{t_{ij}i_j}^i \cdot a_{ij}} \quad (5.13)$$

Siendo  $t_{ij}$  el tiempo que tarda el agente "i" en realizar el estado o tarea "j".

Siendo  $t_{ij} = \text{Máx}_k \{t_{ijk}\}$  dónde  $t_{ijk}$  indica el tiempo que tarda el agente planificador "i", en realizar la tarea "j" para el agente "k"; tomando el máximo sobre "k" (tipo de turista), nos aseguramos que al agente planificador le de tiempo de llevar a cabo la tarea que tiene que hacer con independencia del tipo de agente. Dichos tiempos, inicialmente se estiman. Para que se entienda mejor, y adelantándonos a la evaluación del mecanismo, en el caso de estudio del capítulo 6, el agente planificador será un guía turístico que que debe calcular rutas y el resto serán agentes turistas, por lo que  $t_{ijk}$  indica el tiempo que tarda el agente guía "i" en realizar la tarea "j" (relacionada con la ruta) para el turista "k".

Ahora pasamos a definir las restricciones del problema.

1.- Queremos que cada estado sea hecho por un agente, lo cual en lenguaje matemático para cada estado "k", se traduce en:

$$\sum_{i=1}^m a_{ik} = 1 \quad \forall k \in \{1, \dots, n\} \quad (5.14)$$

2.- Queremos que cada estado se haga dentro de un límite de tiempo. Supongamos que el estado "k" se ha de hacer como mucho en un tiempo  $t_k$ . La restricción sería:

$$\sum_{i=1}^m t_{ik} a_{ik} \leq t_k \quad \forall k \in \{1, \dots, n\} \quad (5.15)$$

3.- Cada estado "k" necesita de una serie de recursos para su ejecución, estos recursos no tienen por qué tenerlos todos los agentes.

Fijado estado "k", necesitaremos  $r_h^k$  recursos con  $h \in \mathbb{N}$ , sea  $r_w = \text{máx}_{h \in \mathbb{N}} \{r_h^k\}_{k=1, \dots, n}$ .

Las variables  $\{r_x^k\}_{x \in \{1, \dots, w\}} \quad \forall k \in \{1, \dots, n\}$  se definen de forma binaria:

$$r_x^k = \begin{cases} 1 & \text{si el agente "k" necesita el recurso "x"} \\ 0 & \text{en otro caso} \end{cases} \quad (5.16)$$

El agente que haga el estado "k" tiene que tener como mínimo a su disposición los recursos que son necesarios para realizar el estado "k", de ahí que fijado estado "k", para cada recurso del conjunto  $\{r_x^k\}_{x \in \{1, \dots, w\}}$   $\forall k \in \{1, \dots, n\}$  se define la siguiente restricción:

$$\sum_{i=1}^m r_{ix} a_{ik} \geq r_x^k \quad \forall k \in \{1, \dots, n\}, \forall x \in \{1, \dots, w\} \quad (5.17)$$

Las variables  $\{r_{ix}\}_{x \in \{1, \dots, w\}}$   $\forall i \in \{1, \dots, m\}$  son variables binarias:

$$r_{ix} = \begin{cases} 1 & \text{si el agente "i" tiene el recurso "x"} \\ 0 & \text{en otro caso} \end{cases} \quad (5.18)$$

4.- Cada agente "i" tiene un tiempo mínimo y máximo de trabajo, el cual depende del tipo de agente, estos tiempos los denoto respectivamente por:

$$t_i^{\text{Turn on}} \text{ and } t_i^{\text{Turn off}}$$

$$t_i^{\text{Turn on}} \leq \sum_{j=1}^n t_{ij} \leq t_i^{\text{Turn off}} \quad \forall i \in \{1, \dots, m\} \quad (5.19)$$

Para la mayoría de los agentes, como veremos en el caso de estudio, el número máximo de horas que puede estar trabajando, se corresponde con una jornada laboral de 8 horas.

5.- Cada vez que asignemos a un agente tareas, queremos que haga un número mínimo de tareas, dicho número depende del tipo de agente.

$$\sum_{j=1}^n a_{ij} \geq \text{NumberTask}_i \quad \forall i \in \{1, \dots, m\} \quad (5.20)$$

Si el problema de programación no lineal planteado fuera incompatible, se añadirían agentes con el fin de que fuera compatible. El agente a añadir es aquél que tuviera la mayor probabilidad a priori de llevar a cabo las tareas a realizar.

Si una norma (restricción) cambia, habría que volver a hacer la asignación de tareas.

De este modo, obtenemos un plan para las tareas que se desean llevar a cabo mediante la organización de agentes. En otras palabras, obtenemos el plan global compuesto por todas las tareas y los agentes de la organización que van a llevarlas a cabo. Cada agente de la organización conoce las tareas que tiene que realizar. Estos agentes, de tipo CBP-BDI integran las 4 etapas de un sistema CBR (recuperación, reutilización, revisión y retención) (ver Figura 12). En cada una de estas etapas la planificación basada en casos realiza las siguientes acciones [Corchado et al., 2008]:

- ❖ Etapa de recuperación. Selección de los planes con información similar al caso problema.
- ❖ Etapa de reutilización. Durante esta etapa el agente crea un espacio vectorial hiper-dimensional que incluye todas las restricciones del problema utilizando una técnica de B-splines. En ese momento se identifican todos los planes que satisfacen los requisitos del usuario. Se trata de curvas geodésicas (en el espacio vectorial hiper-dimensional), que se pueden calcular utilizando el teorema de Euler. Los planes geodésicos garantizan mínimo riesgo y eficiencia constante, y satisfacen los objetivos del agente. Aquel de los planes geodésicos con un campo de Jacobi mínimo es el plan más fácilmente replanificable. El campo de Jacobi mínimo identifica al plan que tiene una mayor densidad de planes a su alrededor, es decir, si el plan se interrumpe, es posible encontrar otro plan cercano que permita alcanzar los objetivos iniciales. El agente CBP-BDI recomienda a sus usuarios los planes que son más fácilmente replanificables.
- ❖ Etapa de revisión. Durante la etapa de revisión el usuario confirma el plan.
- ❖ Etapa de retención. Durante la etapa de retención los planes se almacenan, dependiendo de la eficiencia obtenida.

De esta forma es posible obtener un planificador enfocado a proporcionar planes óptimos, con la mayor eficiencia posible y con el mínimo riesgo en caso de interrupción para un agente CBP-BDI. Se considera un problema de planificación sujeto a ciertas restricciones.

Si para resolver un problema de planificación con restricciones se parte de un conjunto de estados o tareas  $E = \{e_0, \dots, e_n\}$  que se deben completar, es posible definir un plan individual como una secuencia de acciones que, partiendo de un estado  $e_0$ , define la secuencia de estados por la que debe pasar un agente de cara a proporcionar una realización óptima de sus tareas:



$$\bar{a}_j : E \xrightarrow{e_{i,j}} E \quad (5.21)$$

La relación dinámica que se establece entre el comportamiento de un agente y los cambios que se producen en el entorno se puede modelar mediante la ecuación siguiente. En ella se indica que el comportamiento de un agente puede representarse por medio de su función de acción  $a_A(t) \forall t$ , que se define como una correspondencia entre un instante de tiempo  $t$  y la acción seleccionada por el agente.

$$\text{Agent } A = \{a_A(t)\}_{t \in T \subseteq N} \quad (5.22)$$

A partir de la definición de función de acción es posible definir una nueva relación que representa la idea de plan de acción para un agente.

$$p_A : Tx A \xrightarrow{(t, a_A(t))} A \quad (5.23)$$

De la siguiente manera:

$$p_A(t_n) = \sum_{i=1}^n a_{iA}(t_i - t_{i-1}) \quad (5.24)$$

De esta forma ha sido posible obtener una expresión de tiempo discreto para un plan de acción para un agente. Sin embargo, el objetivo del agente CBP-BDI es el de obtener un sistema de planificación dinámico. Así pues, para obtener la definición de plan para un agente CBP-BDI, se presenta la extensión para tiempo continuo de la expresión propuesta en la ecuación, en otras palabras:

$$p_A(t_n) = \int_{t_0}^{t_n} a_A(t) dt \quad (5.25)$$

El entorno en el que se ejecuta el plan es dinámico y el plan de un agente  $p_A(t)$  puede variar. Además, según el modelo CBP-BDI, un plan también podrá ser modificado en función del conocimiento de éxito o fracaso de planes que fueron ejecutados anteriormente. Todos los planes llevan asociada una función de eficiencia definida por una métrica que permite valorar la bondad de un plan de acuerdo a sus objetivos iniciales, a los recursos utilizados para obtener dichos objetivos y a los resultados obtenidos tras la ejecución del plan. La expresión para la eficiencia de un plan se define como la relación entre los objetivos alcanzados y los recursos consumidos.  $O$  indica los objetivos iniciales del plan y  $O'$  representa los objetivos alcanzados tras la ejecución del plan.  $R$  es el total de recursos disponibles para la ejecución del plan y  $R'$  son los recursos

consumidos por el agente durante la ejecución del plan. El operador # representa el cardinal de un conjunto.

$$E_{ff} = \frac{\#(O' \cap O)}{\#R'} \quad (5.26)$$

El agente planificador debe proporcionar planes de eficiencia óptima, pero con mínimo riesgo. Esto es, debe buscar entre los planes de eficiencia óptima aquellos con más posibilidades de replanificación en el caso de que se produzca una interrupción [Corchado et al., 2008].

Dado un estado inicial  $e_0$  de un agente, el término problema de *planificación individual* se utiliza para describir la búsqueda de una solución que permita alcanzar un estado final  $e_i \equiv e^* \in E$  que cumpla con una serie de requisitos. En este momento tenemos el plan  $p^*(t)$  que permite a un agente alcanzar de una manera óptima sus objetivos y subsistir en un entorno dinámico.  $p^*(t)$  proporciona todos los estados por los que tiene que pasar un agente para llevar a cabo una solución establecida como óptima para llevar a cabo las tareas definidas, teniendo en cuenta los recursos disponibles [Corchado et al., 2008]. Se obtiene una estructura para un agente planificador CBP-BDI con un comportamiento dinámico, esto es, el agente tiene en cuenta la posibilidad de que ocurran cambios durante la ejecución de un plan. El agente CBP-BDI representa el problema en un espacio de planificación que estará acotado por las restricciones impuestas por los requisitos. El espacio de planificación ayuda a comprender el problema de planificación para un agente y puede representarse de la siguiente manera:

Tenemos una variable discreta,  $X$ , que toma valores en un conjunto y su variable acumulada  $Ac(X)$  que se construye asignando a cada uno de los posibles valores  $x_i$  que toma la variable  $X$ , el total de resultados previos.

$$X = \{x_i\}_{i \in N} \quad (5.27)$$

Si  $X$  es discreta, el valor  $i^{\text{th}}$  de la variable  $Ac(X)$  se define como:

$$Ac(x_i) = \sum_{j=1}^i x_j \quad \forall x_i \in X \quad (5.28)$$

Si la variable  $X$  es continua y toma valores en el intervalo  $[a,b]$ , quedará representada por medio de la función  $x(t)$ , y la variable  $Ac(X)$  en el punto  $x_i \in [a,b]$  se define como:

$$Ac(x_i) = \int_a^{x_i} x(t)dt \quad \forall x_i \in [a, b] \quad (5.29)$$

Dado un problema  $E$  y un plan  $p(t)$ , es posible construir las funciones  $Ob$  y  $Rc$  acumuladas para los objetivos y costes de dicho plan. De esta forma, para todos los instantes de tiempo  $t_i$  es posible definir dos variables:

$$Ob(t_i) = \int_a^{t_i} O(t)dt \quad (5.30)$$

$$Rc(t_i) = \int_a^{t_i} R(t)dt$$

De esta forma es posible construir un espacio que representa el entorno de un problema de planificación como un espacio hyper-dimensional donde los ejes representan las variables acumuladas de los objetivos y los recursos asociados. El espacio de planificación construido de esta forma cumple las siguientes propiedades:

1. Las representaciones de los planes en el espacio de planificación son siempre funciones monótonamente crecientes. Dado que  $Ob(t)$  y  $Rc(t)$  son funciones definidas como positivas, la función  $p(t)$  expresada en estas coordenadas es constante o es creciente.
2. En el espacio de planificación, las líneas rectas representan planes de eficiencia constante. Si las representaciones de los planes son líneas rectas, la pendiente de las funciones es constante, y coincide con la definición de eficiencia del plan.

$$\frac{d}{dt} p(t) = cte \Leftrightarrow \lim_{\Delta \rightarrow 0} \frac{\Delta O(t)}{\Delta R(t)} = cte \quad (5.31)$$

En un espacio n-dimensional, la extensión del concepto de línea recta se denomina curva geodésica. Es la línea de mínima longitud que une dos puntos en una superficie dada, y está contenida en esta superficie. Las geodésicas de una superficie son las líneas "más rectas" posibles (con menor curvatura) fijado un punto y una dirección dada sobre dicha superficie. En este sentido, es posible introducir la noción de plan geodésico, que se define como un plan que mantiene su eficiencia constante a lo largo de su desarrollo.

Quizás sea posible entender mejor el concepto de plan geodésico si se entiende como la idea de "plan de mínimo riesgo". De esta forma, dado un problema, el agente:

1. Busca el plan que proporcionará una solución conforme a una serie de restricciones  $F(O;R)=0$ .
2. Recupera de la base de planes aquellos planes que son inicialmente compatibles con el problema que se intenta resolver, con los

requisitos impuestos sobre la solución de acuerdo a los deseos del agente y con el estado actual [Aamodt y Plaza, 1994].

3. Representa todos los posibles planes  $\{p_1, \dots, p_n\}$  en el espacio de planificación, obteniendo un subconjunto de estados que el agente ya ha alcanzado anteriormente en el pasado para resolver problemas similares.

Así pues, se obtiene una representación que consiste en una malla de puntos (generalmente irregular) sobre el espacio de planificación. Si se utilizan técnicas de planificación, entonces es posible obtener un hiperplano  $h(x)$  (que encapsula la información correspondiente a experiencias pasadas similares en el conjunto de restricciones impuesto, dando lugar por definición a un hiperplano debido a que verifica que  $h(x_j)=p_j$   $j=1, \dots, n$  y a que el espacio de planificación es de dimensión  $n$ ). Sobre el hiperplano es posible calcular planes geodésicos, sobre los que es posible aplicar cálculo variacional.

Supongamos, por simplicidad un espacio de planificación de dimensión 3 con coordenadas  $\{O, R_1, R_2\}$ . El teorema de Euler [Jost y Li-Jost, 1998] garantiza que si entre el punto  $e_0$  y los puntos objetivo  $f_s f=\{e_1, \dots, e_m\}$  y sobre la superficie de interpolación, la expresión para los planes geodésicos podrá ser obtenida si se resuelve el sistema de ecuaciones presentado en la ecuación (32):

$$\begin{cases} \frac{\partial L}{\partial R_1} - \frac{d}{dO} \frac{\partial L}{\partial R_1'} = 0 \\ \frac{\partial L}{\partial R_2} - \frac{d}{dO} \frac{\partial L}{\partial R_2'} = 0 \end{cases} \quad (5.32)$$

Donde  $R_i$  es la función acumulada de los recursos  $R$ ,  $O$  es la función acumulada del conjunto de objetivos  $O$  y  $L$  es la función de distancia utilizada sobre el hiperplano  $h(x)$ ,

$$L = \int_h dl \quad (5.33)$$

De cara a obtener todos los planes geodésicos que sobre la superficie  $h(x)$  comenzando en el punto  $e_0$ , permiten alcanzar alguno de los puntos  $e \in f_s f$ , es necesario imponer una condición sobre el entorno: el punto inicial debe ser  $e_0=(O_0, R_0)$ .

Utilizando técnicas de cálculo variacional es posible obtener expresiones para todos los aquellos planes que comenzando en  $e_0$  permiten alcanzar los puntos objetivos. Una vez que se han obtenido los planes geodésicos que permiten construir soluciones eficientes que permiten pasar desde el estado actual hasta el conjunto de estados solución, será posible calcular el plan óptimo para dicha trayectoria por medio de una función de distribución de densidad de planes

geodésicos (se elige aquel plan que tiene el mayor número de planes geodésicos a su alrededor, el plan con menor distancia al resto de planes). La herramienta que posibilita la realización de este cálculo se denomina campo de Jacobi mínimo asociado con el conjunto solución.

Si  $g_0: [0,1] \rightarrow S$  es una geodésica sobre la superficie  $S$ .  $h: [0,1] \times [-\varepsilon, \varepsilon] \rightarrow S$  es una variación de  $g_0$  tal que para cada  $t \in (-\varepsilon, \varepsilon)$ , el conjunto  $\{h_t(s)\}_{t \in (-\varepsilon, \varepsilon)}$ :  $h_t(s)$  para todo  $t \in (-\varepsilon, \varepsilon)$  son geodésicas en  $S$  que comienzan en  $g_0(0)$ , en otras palabras, cumplen con que  $h_t(0) = g_0(0)$  para todo  $t \in (-\varepsilon, \varepsilon)$ . Bajo estas condiciones, llevando las variaciones hasta un límite diferencial, es posible obtener la ecuación (34):

$$\lim_{t \rightarrow 0} \{h_t(s) = g_0(s+t)\} = \lim_{t \rightarrow 0} \{h(s,t)\} = \left. \frac{\partial g_0}{\partial t} \right|_{(s,0)} = \frac{dg_0}{ds} \equiv J_{g_0}(s) \quad (5.34)$$

El término  $J_{g_0}(s)$  se aplica al campo de Jacobi de la geodésica  $g_0$  para el conjunto  $\{g_n(x)\}_{n \in N}$ , y de la misma forma se ha construido la definición de  $J_{g_0}(s)$  es posible proporcionar una medida para la distribución del resto de geodésicas  $\{g_n(x)\}_{n \in N}$  alrededor de  $g_0$  a lo largo de la trayectoria. Dado un conjunto de geodésicas, algunas de ellas son siempre  $g^*$  que, en su entorno, tienen una mayor distribución que otras geodésicas en su entorno cercano. Esto es equivalente a decir que presentan una variación en la distribución de geodésicas menor que el resto y además el campo de Jacobi asociado con  $\{g_n(x)\}_{n \in N}$  alcanza su valor más bajo en  $J_{g^*}$ .

Volviendo al problema de identificar aquella intención que sea más replanificable, teniendo en cuenta que en el CBP se realizan cuatro etapas secuenciales, si se aplica recuperación de experiencias pasadas similares y técnicas de calculo variacional, se obtiene un conjunto de planes geodésicos  $\{p_1, \dots, p_n\}$ . Si se selecciona al plan óptimo  $p^*$  por medio de un valor mínimo del campo de Jacobi, es posible garantizar que si se produce una interrupción, el plan tendrá el mayor número de planes geodésicos a su alrededor de cara a resolver la interrupción y poder continuar con la ejecución. Esto sugiere que dado un problema con una serie de restricciones  $F(O;R)=0$ , el plan geodésico  $p^*$  con un campo de Jacobi mínimo asociado con el conjunto  $\{g_n(x)\}_{n \in N}$  puede ser denominado solución más replanificable. De esta forma, el modelo de comportamiento  $G$  para el agente CBP-BDI viene dado por medio de la ecuación:

$$G(e_0, p_1, \dots, p_n) = p^* \Leftrightarrow \exists n \in N / J_{g_n} \equiv J_{g^*} = \underset{n \in N}{\text{Min}} J_{g_n} \quad (5.35)$$

Si el plan  $p^*$  no se ve interrumpido, el agente alcanzará un estado objetivo  $e_j \equiv e^* \in f_s, f, j \in \{1, \dots, m\}$ . Cuando el plan termina su ejecución por alcanzar el estado final, es necesario realizar un aprendizaje. En la fase de aprendizaje se almacena un peso  $w_f(p)$  que contiene la información relativa a la eficiencia del

plan. Con la actualización de los pesos  $w_f(p^*)$ , es posible completar el ciclo de planificación CBP.

Por el contrario, si el agente inicia un  $p^*$  pero en un instante de tiempo  $t > t_0$  el plan sufre una interrupción (a causa de un cambio en su entorno), entonces será necesario efectuar una replanificación, tal y como se puede apreciar en la Figura 11. La planificación geodésica cumple con las condiciones del principio de optimalidad de Bellman [Bellman, 1957], esto es, cada uno de los sub-planes (planes parciales) es parcialmente geodésico entre los puntos seleccionados. Esto garantiza que si  $g_0$  es geodésico, para una interrupción de  $e_0$  en el instante de tiempo  $t_1$ , dado que  $e_0$  cambia a  $e_1$ , y que  $g_1$  es geodésico a  $e_1$  que comienza en el estado en el que  $g_0$  fue interrumpido, entonces se cumple que  $g = g_0 + g_1$  es geodésico a  $e = e_0(t_1 - t_0) + e_1(t_2 - t_1)$ . El proceso dinámico se realiza por medio del ciclo CBP de forma recursiva: cada vez que un plan es interrumpido se toma el último estado alcanzado como estado inicial, se analizan las nuevas condiciones del entorno, y se recuperan nuevos planes de la memoria de planes que se ajustan al nuevo problema. A partir de este punto, se calculan los planes geodésicos y de entre ellos se selecciona aquel que cumpla las condiciones mínimas del campo de Jacobi asociado.

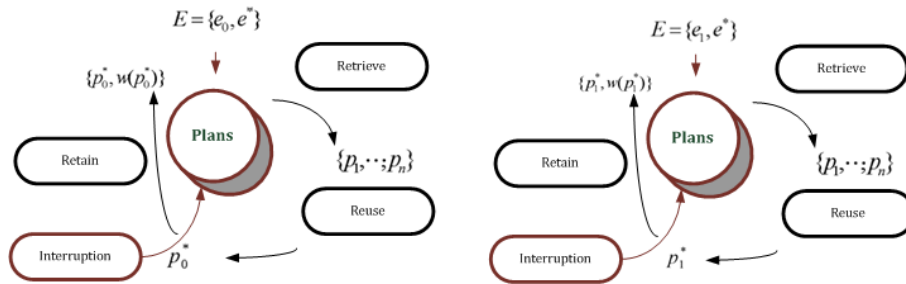


Figura 11. Representación de la actuación del agente CBP-BDI ante una situación de interrupción

Un campo de Jacobi mínimo global  $J(t)$  también cumple con las condiciones de optimalidad de Bellman [Bellman, 1957], es decir, un campo de Jacobi mínimo global está formado por campos de Jacobi mínimos en piezas.

$$J_{\min}(t) = \{J_{\min}(t_1 - t_0), J_{\min}(t_2 - t_1), \dots, J_{\min}(t_n - t_{n-1})\} \quad (5.36)$$

Si por un lado, sucesivos campos de Jacobi generan un campo de Jacobi global, y por otro lado campos de Jacobi mínimos generan un campo de Jacobi mínimo, el agente CBP-BDI tiene una estrategia de replanificación  $G(t)$  de la forma que se ha indicado anteriormente para sobrevivir en un entorno dinámico [Bajo et al., 2007], [Bajo et al., 2008], [Corchado et al., 2007], [Corchado et al., 2008].

El agente CBP-BDI posee capacidades de aprendizaje y adaptación. Esto hace que resulte adecuado para la resolución de problemas en entornos dinámicos. El agente, de manera individual, partiendo de una base de planes, puede diseñar y re-diseñar, estrategias versátiles si sus planes en curso son interrumpidos, sin necesidad de que este proceso afecte al resto de agentes de la organización. Pero hay que tener en cuenta que los problemas a resolver y los objetivos a cumplir vendrán dados a la organización, a la sociedad de agentes, por lo que inicialmente, al introducir un problema a resolver en la organización habrá que considerar las restricciones de la misma. Por ejemplo, solo determinados agentes podrán llevar a cabo planes que lleguen a una solución óptima. Una vez identificados estos agentes, habrá que asignar las acciones que llevará a cabo cada agente, es decir, generar el *plan global* en el que se indicará los estados por los que se pasará para llegar a la solución. Habrá que definir qué agente será el encargado de pasar de un estado  $e_i$  a  $e_j$ . Y tendrán que coordinarse para llevar a cabo el plan global. También se deberán tener en cuenta las normas que indican la forma de comunicarse los agentes entre sí (obligaciones, permisos, etc.), definidas en la arquitectura. Así como los servicios que pueden llevar a cabo cada agente y con qué rol.

Según el modelo planteado, en un momento concreto  $t$  de la vida de la organización pueden pasar tres tipos de situaciones:

1. Si un agente planificador tiene un problema, debe llevar a cabo una replanificación de manera individual. En caso de no ser posible (no puede llevar a cabo las tareas encomendadas por el coordinador), se procederá a una reasignación global de tareas, en las que podrán variar las tareas asignadas a cada agente planificador.
2. Si cambia la estructura de la sociedad, por ejemplo, entra un nuevo agente a formar parte de la sociedad, se deberá realizar una reasignación global de manera que se replanifiquen todos los agentes.
3. Lo mismo ocurrirá si hay un cambio de normas. Se tendrá que realizar una reasignación a nivel de organización.

Estas situaciones, que pueden verse de manera gráfica en la Figura 10, harán que el sistema deba adaptarse de manera dinámica al plan establecido inicialmente, variando para ello las tareas que debe realizar la organización. Se pretende minimizar en lo posible los cambios en los objetivos iniciales, y por ello, se intenta primero realizar una planificación a nivel individual de cada agente. Esto será sencillo ya que el plan elegido inicialmente por cada agente planificador es el óptimo, es decir, el plan más replanificable en caso de fallo. Después, en caso de que esto no sea posible, se realizará una reasignación de manera global. En el siguiente capítulo se expondrá un caso de estudio en el que se pondrá a prueba el modelo definido. Para ello se dispondrá una organización de agentes especializada en ciertas tareas (relacionadas con el turismo) y una planificación que tiene que llevarse a cabo mediante los agentes que la forman. Será necesario aplicar el modelo definido para realizar ese plan y para verificar

que es posible una replanificación en caso de que ocurra algún contratiempo en tiempo de ejecución

Un esquema del funcionamiento general del sistema, enfocado ya al caso de estudio que se describirá más adelante, puede verse en la siguiente figura. En el caso de estudio, los agentes CBP-BDI, los agentes que llevarán a cabo su planificación individual serán los agentes con el rol de "Guía". Y el agente con el rol de "Coordinador" será el encargado de la planificación global. Además habrá otros roles dentro de la organización como "Turistas", "Monumentos", etc. Por un lado, tendremos un sistema CBR en el que se coordinarán las tareas de la organización y en el que se almacenarán las tareas llevadas a cabo por todos los agentes. Por otro lado, al igual que se planteaba en [Corchado et al. , 2008], tendremos para cada agente guía CBP-BDI, un ciclo en el que cada agente obtendrá su plan óptimo de manera individual.

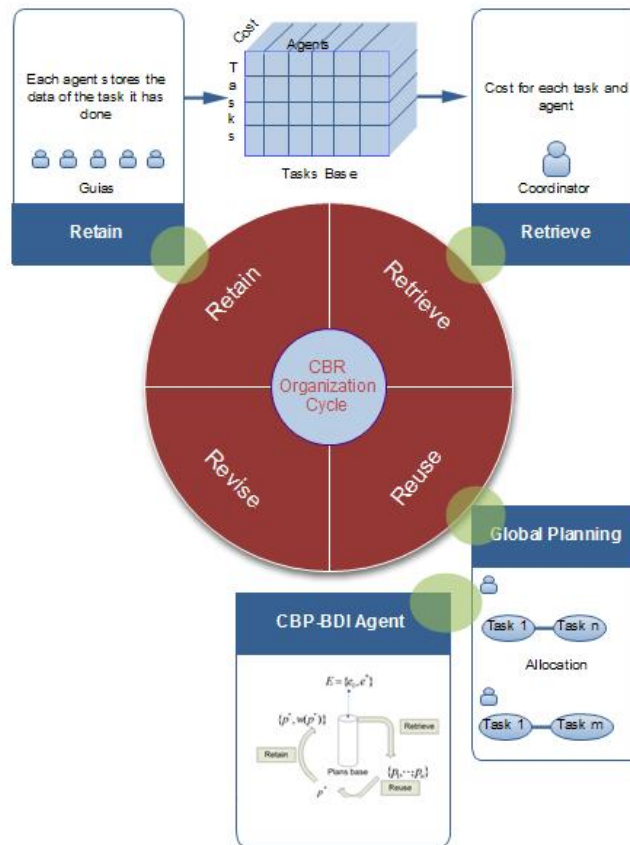


Figura 12. Esquema de funcionamiento del sistema



El coordinador recupera de la base de tareas los agentes candidatos a llevar a cabo las tareas que conforman el plan de la organización y asigna a cada agente planificador las más adecuadas según su perfil. Cada uno de los agentes planificadores, de tipo CBP-BDI genera su plan óptimo y empieza ejecutar sus actividades. Tienen la capacidad de adaptarse en cada paso de su plan, así que si surgiera un comportamiento emergente, pueden replanificarse. Durante su etapa de retención recolecta sus planes, dependiendo de la eficiencia obtenida. Además, cada uno de los agentes almacena los datos de las tareas que ha realizado y que le fueron asignadas desde la coordinación inicial, para que en futuros planes de la organización sean tenidas en cuenta sus acciones.

---

#### *5.4 CONCLUSIONES PRELIMINARES*

---

Las razones propuestas por Ferber [Ferber, 1999] para la realización de acciones coordinadas que veíamos en la sección 2.2.1 se sustentan ahora, haciendo que los agentes puedan compartir recursos, que se optimice la eficiencia del sistema y que los agentes puedan alcanzar sus objetivos y adaptarse de una manera independiente.

El modelo social de planificación propuesto se presenta entre los pocos que ofrecen la posibilidad de comportamientos deliberativos y sociales. Cabe destacar que es un modelo único ya que incorpora un mecanismo propio de reorganización y adaptación social. THOMAS facilita el desarrollo de SMA en un paradigma organizacional y el modelo propuesto añade las funcionalidades de reorganización y adaptación.



PARTE IV. EVALUACIÓN Y  
CONCLUSIONES



---

## 6 CASO DE ESTUDIO

---

De cara a evaluar el modelo propuesto en este trabajo de tesis, se realizan una serie de experimentos. Para ello se dispone de tecnología para el desarrollo de organizaciones virtuales de agentes en entornos abiertos, en concreto, de la arquitectura THOMAS [Giret et al., 2009][Carrascosa et al., 2009]. Las funcionalidades de reorganización y adaptación de los agentes en su comportamiento son necesarias para esta plataforma. Para ello se ha propuesto el modelo detallado en el capítulo anterior y que será validado mediante el siguiente caso de estudio.

Los objetivos a alcanzar en el caso de estudio son:

- ❖ Desarrollo de la sociedad de agentes lo que incluye:
  - Definición de roles de agentes: se definirán las funcionalidades generales necesarias para el correcto funcionamiento de la sociedad y las propias del caso de estudio definido. De esta forma quedarán determinados los comportamientos de los agentes que formarán la organización. Hay que tener en cuenta que el sistema será abierto por lo que la definición de estos roles no debe ser condición de imposibilidad para que puedan utilizarse sistemas externos sobre el mismo.
  - Definición de los servicios necesarios para el adecuado funcionamiento de una organización.
  - Definición de las normas que regirán la sociedad y que influirán en las interacciones y comportamiento de los agentes.
  - Definición de los mensajes e interacciones propios de los agentes.
- ❖ Simulación del comportamiento de la organización en un caso concreto de coordinación y adaptación de sus agentes.
- ❖ Validación del modelo propuesto mediante la simulación de la organización del caso de estudio.

El capítulo se ha dividido de la siguiente forma: en primer lugar se describirá una introducción al caso de estudio, presentando las motivaciones que llevaron a elegir este campo en concreto y algunos trabajos relacionados existentes sobre el tema. En segundo lugar, se describe el contexto en el que se desarrolla el caso de estudio, presentando el funcionamiento del sistema tanto de forma interna (funcionamiento general del modelo) como de forma externa (agentes que existen en el mundo). La sección 6.3 mostrará el sistema diseñado, *TouristOrg*. Por último, en la sección 6.4 se presenta un ejemplo concreto de adaptación de los planes propuestos para la organización. Las conclusiones obtenidas durante el desarrollo del caso de estudio serán descritas junto con los resultados obtenidos en el siguiente capítulo.

## 6.1 INTRODUCCIÓN

Los servicios basados en las nuevas tecnologías, están generando nuevas líneas de investigación en todo lo referente a la mejora de la experiencia de los usuarios, eliminando los servicios irrelevantes o de falta de interés y ajustándose a las verdaderas necesidades de cada uno. El sector turístico ha sido de los primeros en utilizar tecnologías de la información. A la vista está la gran diversidad de sistemas de información turística que encontramos, tanto en las aplicaciones como en las tecnologías utilizadas para su implementación (sección 6.1.1).

La diversidad de los datos y la interoperabilidad son dos desafíos conocidos en el desarrollo del mercado turístico electrónico, en el que las empresas necesitan intercambiar información de una manera transparente. El nuevo turista tiene unas motivaciones y unas necesidades, unos hábitos de información, de compra y de utilización, distintos a los conocidos no hace muchos años. La Figura 13 muestra un esquema de líneas de actuación de las nuevas tecnologías sobre el sector turístico.

Una de las principales tendencias en este sector, es implementar sistemas para ayudar a los turistas en la planificación de sus rutas. Los sistemas de recomendación y guiado son un intento de modelizar matemáticamente y reproducir técnicamente el proceso de recomendación en el mundo real. Se trata de aplicaciones reconocidas de las técnicas de Inteligencia Artificial basadas en componentes software típicos en el e-comercio y sistemas turísticos.

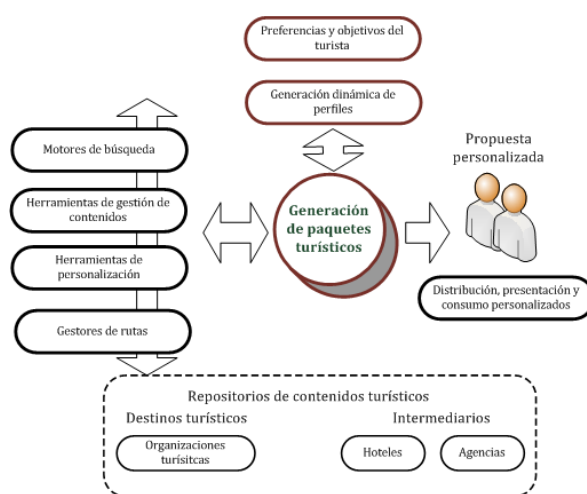


Figura 13. Nuevas tecnologías en el sector turístico

Dentro de las técnicas de IA empleadas destacan los SMA (sistemas multi-agente). Los SMA han sido probados satisfactoriamente en aplicaciones de este tipo [Fonseca et al., 2001][Kowalczyk et al., 2002] [Bajo et al., 2009]. Sus características de autonomía, flexibilidad y capacidad de cooperación, los hacen adecuados para ambientes dinámicos como éstos. Por su parte, la tendencia actual de estos sistemas a emplear conceptos organizacionales en su diseño, ayuda a enfocar su empleo en el campo del turismo, ya que proporciona el grado de emparejamiento propicio entre organización turística y organización a nivel de agentes software respecto a estructura, funcionalidad, etc. Las organizaciones virtuales [Ferber et al., 2004] son un medio de entender el modelo de sistemas desde un punto de vista sociológico muy acorde al área en cuestión. El modelo de las organizaciones virtuales desde un punto de vista empresarial se apoya en los principios de la cooperación entre empresas dentro de una red, y explota aquellos elementos diferenciadores que aportan flexibilidad, capacidad y rapidez de respuesta, dentro de una estrategia dirigida a satisfacer al consumidor [Schertler, 1998]. Resulta pues adecuado que las empresas turísticas vayan tomando en consideración esta forma de organización que explota de manera eficiente las nuevas tecnologías de la información y la comunicación, para mejorar la competitividad de las empresas turísticas.

Aún así, dentro del desarrollo de organizaciones, tanto a nivel empresarial como a nivel de agentes, hemos visto como existen un conjunto de requerimientos [Rodríguez et al., 2009] que hacen necesarios nuevos modelos sociales en los que se permita la utilización de sistemas abiertos [Zambonelli et al., 2003] y adaptativos y adaptativos [Zambonelli et al., 2004][Reitbauer et al., 2004][Weyns et al., 2004] [Capera et al., 2003][Razavi et al., 2005].

Incluso centrándonos en el ámbito turístico podemos ver que la adaptación se considera un denominador común en las investigaciones en este área. Adaptación de:

- ❖ la oferta (un contexto turístico concreto y a unos requerimientos de servicios básicos para realizar la actividad);
- ❖ los proveedores (la información turística y la base de datos de rutas);
- ❖ y los consumidores/usuarios (los modelos de usuarios).

Todos estos factores son clave a la hora de personalizar una ruta adaptada al servicio que se quiere ofertar. Es innegable que la efectividad de las diferentes técnicas va a estar en función del producto o servicio al que se apliquen, ya que estarán integradas en un sistema directamente relacionado con el conocimiento de los usuarios. En el sector tradicional o lo que es lo mismo, en las oficinas de turismo clásicas, esto simplemente puede envolver una conversación entre la persona que atiende al público y un cliente. Ahora bien, la situación a través de un medio técnico que determina los productos o tipo de productos que interesan a un cliente específico es diferente, ya que se pueden emplear

diferentes métodos y herramientas para proporcionar un servicio personalizado. En estos casos, la interacción debe ser constante a través de estrategias de control personalizadas: sistemas de recomendación, gestión de datos, instrumentos de software de comunicación con el cliente, es decir todo aquello que suministra al servicio de información sobre el mismo, siendo el objetivo ofertar ventajas (como valor añadido) ante posibles competidores.

Por lo tanto, los turistas son la parte esencial del sistema. Los turistas que están motivados a realizar una ruta turística deben elegir a dónde ir, cómo llegar al destino y qué hacer en dicho destino. El proceso de planificación de las rutas es complejo, ya que existe un gran número de factores en dicho proceso como las preferencias de los turistas por determinadas atracciones, el medio de transporte, los servicios, y las limitaciones de tiempo y dinero.

- ❖ Motivaciones y preferencias de los turistas. Cualquier viaje debe satisfacer en primer lugar las motivaciones y las preferencias de los turistas. Estos factores influyen sobre la decisión de visitar las atracciones, el alojamiento y el medio de transporte. La imagen de las atracciones, sus horarios de apertura y cierre, y los servicios ofertados son elementos clave que influyen a los turistas a la hora de elegir los destinos. Adicionalmente, la decisión también está influenciada por otros parámetros más subjetivos como la salud, hobbies, religión y cultura.
- ❖ Recursos de los turistas. Además de las preferencias, las restricciones de tiempo y dinero también tienen un gran impacto en las decisiones de los turistas, ya que sólo se eligen los destinos y atracciones que se correspondan con las limitaciones económicas. Casi todos los turistas disponen de un tiempo limitado que quieren maximizar, por lo que es importante conocer datos como el tiempo de visita a determinadas atracciones, los momentos del día más idóneos para hacerlo o el tiempo de desplazamiento del alojamiento a las atracciones o entre las atracciones entre sí.

Cuando un turista define su agenda, elige una serie de actividades que quiere realizar durante el día, teniendo en cuenta no sólo sus intereses, sino también una serie de restricciones. En primer lugar, existen restricciones de tiempo. Aunque una agenda turística no es un horario estricto, suele incluir información sobre las actividades que se van a realizar por la mañana, por la tarde o por la noche. Es evidente que las actividades programadas para cada periodo del día no pueden superar el tiempo disponible.

Un segundo tipo de restricciones está relacionado con los periodos en los que se pueden realizar las actividades planificadas. En este caso, el término periodo tiene varias acepciones: un intervalo de algunos días (monumentos en restauración), un conjunto de días fijos durante la semana (cierres de museos en lunes) o un intervalo de tiempo durante un día (cerrado por la noche). En



tercer lugar, aparecen restricciones debidas a la localización física donde tienen lugar las actividades.

Por ejemplo, no es razonable planificar la visita a cuatro monumentos históricos en una gran ciudad que se encuentran muy alejados unos de otros durante una mañana. Aunque este conjunto de restricciones para la organización de una agenda turística puede ser bastante complejo, no es complicado de entender para el turista. Sin embargo, la realización manual de esta agenda puede resultar un proceso aburrido y hace falta dedicar un tiempo a acceder a las diferentes fuentes de información y realizar los itinerarios.

El proceso de creación de rutas o itinerarios personalizados basados en el perfil de un usuario y en ciertas restricciones, se puede resolver mediante problemas estándares ya conocidos de la Investigación Operativa (IO). IO es una ciencia interdisciplinaria que aplica los métodos científicos para resolver problemas cotidianos dentro de una organización, buscando mejorar la eficiencia. Algunos de los algoritmos asociados a los procesos de planificación son los basados en el problema del viajante (TSP), dichos algoritmos no incorporan ninguna restricción crítica para el sistema turístico funcional (*Travelling Salesman Problem* [Held and Karp, 2005], *Traveling Salesman Problem with Time Windows* [Dumas et al., 1995], *Generalized Travelling Salesman Problem* (GTSP)[Labordere, 1969] *Orienteering Problem* (OP) [Golden et al., 1987], *Chinese Postman Problem* (CPP) [Thimbleby, 2003]). Todos estas soluciones se centran en los aspectos relacionados con la movilidad. Su objetivo es encontrar una ruta entre los puntos de comienzo y final para maximizar un beneficio concreto o minimizar un coste en un tiempo concreto para un sólo usuario. Aunque estas soluciones se han estudiado desde hace bastante tiempo, todavía no existe una solución óptima para los problemas de planificación de rutas escalables a entornos no deterministas y compartidos por varios miembros de una organización. Estos problemas presentan un gran cálculo computacional y no existe una solución que encuentre una planificación auto-adaptada a la variación de puntos en la ruta.

Algunos de estos algoritmos existentes se pueden aplicar al cálculo de rutas en el sector turístico. Las rutas se personalizan basándose en el perfil del turista, las atracciones y la información del viaje. Cada ruta suele comenzar con la planificación del día de turismo de una manera general. Esta fase es importante, pero sería mucho más adecuado crear un itinerario dinámico en el que el sistema ofrece igualmente una lista de las atracciones recomendadas en función de los intereses del usuario, del resto de usuarios y de la posible replanificación de esos planes. Este problema de planificación adaptativa dentro de una organización es extrapolable a otros entornos y casos de estudio cotidianos (entornos empresariales, educativos, de entretenimiento, etc.).

Hemos visto cómo los agentes CBP-BDI nos proporcionan la integración de mecanismos de razonamiento y planificación que permiten llevar a cabo

aspectos de adaptación y reorganización. También hemos examinado las principales técnicas que tratan de resolver el problema de encontrar una secuencia de estados de transición desde un punto (estado) de partida a un punto (estado) de destino, encontrando una solución óptima que resuelva este problema dependiendo del propósito que se persiga. Y por último se ha definido un modelo con el que se pretende llevar a cabo una planificación adaptativa dentro de la organización de agentes. Ahora, vamos a ver de manera experimental todo lo planteado.

Mediante la combinación de tecnología multi-agente y de técnicas de planificación basadas en IO vistas en el modelo propuesto (capítulo 5), se pretende llevar a cabo una distribución óptima de las tareas de los agentes miembros de la organización. Para llevar a cabo su evaluación, se propone un caso de estudio dentro del sector turístico, en el que la generación de planes y la distribución de recursos de forma óptima siempre ha sido un reto conocido. En la aportación de este trabajo tiene un peso importante el caso de estudio diseñado. Sin embargo, el objetivo del mismo no es un estudio del comportamiento de un entorno turístico, sino la validación del modelo de planificación propuesto mediante la simulación de una organización dentro de este sector.

---

#### 6.1.1 *Algunos trabajos relacionados*

---

En la actualidad, existe un gran número de proyectos que se centran en la generación de rutas, proporcionando al turista un soporte dinámico basado en el contexto durante la visita. Un proyecto muy significativo en este ámbito es GUIDE<sup>13</sup>, una guía turística basada en el contexto para los turistas de la ciudad de Lancaster. Este prototipo es capaz de definir una ruta personalizada a pie por la ciudad, facilitar las instrucciones para llegar de un punto a otro y ofrecer información sobre los recursos turísticos. CRUMPET<sup>14</sup> es un proyecto europeo que trata de dar un apoyo móvil a los turistas de diferentes ciudades europeas. Además de un mapa interactivo para crear un itinerario dinámico, el sistema ofrece igualmente una lista de las atracciones recomendadas en función de los intereses del usuario. Otro proyecto conocido relacionado con el tema es Deep Map<sup>15</sup>, que está desarrollando un sistema móvil capaz de generar paseos personalizados guiados por la ciudad de Heidelberg sobre la base de algoritmos de generación de rutas personalizadas por la ciudad. Todos los proyectos mencionados anteriormente se centran en los aspectos relacionados con la

---

<sup>13</sup> <http://www.guide.lancs.ac.uk> [Último acceso 20/1/2010]

<sup>14</sup> <http://www.eml-development.de/english/research/crumpet/index.php> [Último acceso 20/1/2010]

<sup>15</sup> <http://www.eml-development.de/english/research/deepmap/index.php> [Último acceso 20/1/2010]

movilidad. Estos sistemas dan soporte al usuario principalmente durante la propia visita y ofrecen información sobre recursos turísticos y servicios disponibles basados en el contexto, pero es el propio usuario el que genera su agenda. Cada día en un viaje suele comenzar con la planificación del día de turismo de una manera general. Esta fase importante, que suele preceder al propio itinerario, no está cubierta por estos sistemas.

Centrándonos en la aplicación de técnicas de IA dentro del sector turístico y de entretenimiento, encontramos gran diversidad tanto en los sistemas de información turística, como en las aplicaciones o en las tecnologías utilizadas para su implementación. Por ejemplo:

CBCF (*Content-Boosted Collaborative Filtering*) [Melville et al., 2002] es una plataforma para la combinación de recomendaciones colaborativas basadas en el contenido. Para recomendar una película a un usuario activo, esta aproximación utiliza predicción basada en el contenido para aumentar los datos de los usuarios mediante la predicción del ranking de las películas a partir de sus perfiles previos y la provisión de sugerencias personalizadas. El ranking varía entre cero y cinco estrellas, con cero indicando un total desacuerdo.

LIBRA (*Learning Intelligent Book Recommending Agent*) [Mooney and Roy, 2000] es un sistema de recomendación de libros que utiliza extracción de información y algoritmos adaptativos sobre la base de categorización de textos. El sistema recomienda libros en función de las preferencias individuales de los usuarios, utilizando una base de datos de información de libros extraída de las páginas web de Amazon.com<sup>16</sup>. Un sistema de información basado en patrones extrae información sobre el título, autores, sinopsis, opiniones publicadas, comentarios de los consumidores o títulos relacionados. El usuario elige y escala un conjunto de libros proporcionando puntuaciones entre 1 y 10 para cada título. Una vez que se ha creado el perfil, dicho perfil se utiliza para predecir el ranking del resto de los libros disponibles.

MRS (*Music Recommendation System*) [Hung-Chen and Arbee, 2005] proporciona un servicio de recomendación de música. El sistema extrae un fragmento representativo para cada elemento MIDI (*Musical Instrument Digital Interface*). Cada fragmento se clasifica en función de seis parámetros como el nivel de graves/agudos. Este método da lugar a perfiles de diferentes intereses de usuarios y su comportamiento a partir del historial de acceso. Se identifican los usuarios con perfiles similares como usuarios relevantes. Para hacer una recomendación a un usuario, se ponderan los grupos de música asociados con los usuarios relevantes. Estas ponderaciones se pueden almacenar como

---

<sup>16</sup> <http://www.amazon.com/> [Último acceso 28/12/2009]

preferencias asociadas. El sistema calcula las diferencias entre los factores de ponderación en cada perfil de usuario y sus preferencias asociadas.

TIP (*Tourist Information Provider*) [Hinze and Voisard, 2003] [Hinze et al., 2005] proporciona recomendaciones para el usuario sobre su localización y sus perfiles de usuario definidos. El sistema proporciona una lista de las atracciones turísticas próximas que se adecuan a las preferencias de usuario, mientras desecha aquellas que no se adecuan al perfil de usuario.

SHOMAS (*Intelligent Guidance and Suggestions in Shopping Centres*) [Bajo et al., 2009] es un sistema multi-agente para el guiado y la recomendación de actividades en centros comerciales. La arquitectura multi-agente incorpora agentes tanto reactivos como deliberativos que pueden tomar decisiones automáticamente. En concreto, pueden proporcionar sugerencias en tiempo de ejecución con la ayuda de planificadores propios. Además, incorpora tecnología *wireless* y RFID (*Radio Frequency Identification*) que permite a los usuarios interactuar con sus dispositivos móviles y utilizar capacidades de localización.

---

## 6.2 CONTEXTO

---

El caso de estudio presenta una sociedad orientada a ofrecer servicios de guiado para turistas. Dicha sociedad se implementa utilizando el modelo propuesto en el capítulo 5 y se representa en un mundo virtual en el cual estarán disponibles los inventarios de un conjunto documental de patrimonio cultural. Los mundos virtuales dan la posibilidad de incorporar numerosas capacidades de interacción entre los elementos del sistema permitiendo realizar de este modo gran número de tareas. La simulación dentro del mundo virtual representará un entorno turístico en el que habitarán guías y turistas y en el que se llevarán a cabo las tareas propias de guía turística cumpliendo unas determinadas normas.

Para establecer el modelo de interacción es necesario analizar las necesidades y expectativas de los usuarios potenciales del mundo virtual propuesto. A partir de este análisis es posible deducir los roles de los usuarios que participan en el mundo virtual y la forma en la que van a intercambiar informaciones. Los roles del caso de estudio identificados son los siguientes:

- ❖ **Turista:** Este agente se comunica constantemente con el agente Coordinador para obtener las rutas planificadas para él dentro de una zona así como el grupo de turistas al que está asignado. Una zona será un área dentro del mundo virtual con ciertas características atractivas turísticamente. Por ejemplo, dentro de una zona podremos tener varios museos, casas históricas, paseos, eventos musicales, etc. El agente turista tendrá un perfil caracterizado por una serie de variables propias (tiempo, dinero disponible, preferencias, etc.).

- ❖ **Monumento:** un agente monumento podrá ser de dos clases: estático (por ejemplo un museo o un edificio) caracterizado por permanecer inmóvil dentro de nuestro mundo o dinámico (por ejemplo una exposición o un concierto) que podrá estar o no en cierto momento dentro de nuestro mundo. Los atributos de cada uno de ellos serán similares.
- ❖ **Guía:** Un agente guía será una especialización del agente turista con más capacidades. En este caso, además de comunicarse constantemente con el agente Coordinador para obtener las rutas planificadas dentro de una zona, le será asignado un número de visitantes ("un grupo de turistas") para llevar a cabo con ellos la ruta planificada. Dentro de su perfil se incluirán atributos como la empresa en la que trabaja, el grupo asignado de turistas y visitantes, la ruta a seguir, el número de guías disponibles (además de él), su especialización en el conocimiento de ciertos monumentos (preferencia de relación con agentes monumento concretos), así como habilidades propias como el conocimiento de un determinado idioma. El agente coordinador será el encargado, dentro de la sociedad de asignar rutas y tareas a los guías disponibles y asignar visitantes a cada uno de ellos.
- ❖ **Visitante:** Un agente visitante también es una especialización del agente turista, pero en este caso con menos capacidades. Un visitante no dispondrá de las capacidades de acceso a través de dispositivos móviles pero tendrá un perfil con su tiempo, dinero y preferencias y podrá ser asignado a un agente guía para realizar una ruta por la zona. Serán una especialización estática de los agentes turistas (no variarán sus datos a lo largo del tiempo).
- ❖ **Coordinador:** un agente que juegue este rol tendrá las capacidades de asignar las rutas a seguir tanto a los agente guía como a los agentes turistas. Tendrá las capacidades de reorganización.
- ❖ **Notificador:** Será el encargado de realizar los envíos de información en el sistema
- ❖ **Manager.** Realiza el control general del sistema. Controla la conexión y desconexión al sistema de los agentes Turistas, Visitantes y Guías.

Un agente Guía será capaz de generar automática y dinámicamente las rutas a seguir por un Turista o por un Guía, independientemente de los demás pero dependiente de los recursos y perfil de cada uno de ellos. Será el agente planificador de la organización propuesta. El agente Coordinador será el encargado de asignar a cada uno de ellos las tareas a realizar según la mejor planificación global. Es decir, existirán ciertas restricciones individuales de cada agente que serán tenidas en cuenta por los agentes Guía para generar las rutas pero será el agente Coordinador el que tenga en cuenta las restricciones de toda la sociedad de agentes para repartir y asignar las tareas a Guías y Turistas. Decir que tanto el agente Guía como el agente Coordinador tendrán

acceso a la información almacenada en el sistema sobre monumentos, perfiles de guías y turistas, etc.

La información almacenada es una base de casos en la que para cada agente se considera la tarea que llevó a cabo y el coste de recursos que generó. Cuando el agente Coordinador debe realizar una asignación, tiene en cuenta la base de casos de todas las tareas y todos los agentes, recuperando aquellos que contienen al menos una de las tareas que se plantean dentro de los objetivos de la organización. De este modo, el agente Coordinador se está asegurando de que la tarea al menos pueda realizarse. En caso de que alguna de las tareas no esté contenida tendremos que mirar las posibilidades de éxito de cada agente con esa tarea para poder llevar a cabo su asignación.

Un esquema del funcionamiento del sistema puede verse en la Figura 12 . Por un lado, tendremos un sistema CBR en el que se coordinarán las tareas de la organización y en el que se almacenarán las tareas llevadas a cabo por todos los agentes. Por otro lado, al igual que se planteaba en [Corchado et al. , 2008], tendremos para cada agente Guía CBP-BDI, un ciclo en el que cada agente obtendrá su plan óptimo de manera individual.

No se trata de un sistema de guiado típico. En los sistemas de guiado clásicos cada guía es asignado a un grupo de turistas y visitantes y realiza una ruta turística asignada en la empresa para este grupo. En nuestro caso de estudio el sistema es mucho más abierto. La planificación inicial, tanto la asignación de grupos de turistas (y visitantes) a guías, como la ruta a seguir, puede variar. En función de los guías disponibles, de las preferencias y situación de los turistas y de los eventos disponibles para realizar una ruta, se realiza una planificación dinámica para un momento concreto; lo que puede provocar que la sociedad de agentes tenga que adaptarse a las nuevas necesidades.

Por ejemplo, en un momento determinado de una ruta en la que se realiza una "visita a un museo" es posible que sea obligatorio formar grupos de turistas de 5 personas para entrar en el museo. Si en un inicio los grupos estaban formados por 10 personas, el sistema debe ser capaz de reasignar los grupos (cambiar la estructura de la organización) y además introducir un nuevo agente que sea capaz de llevar a cabo el rol de Guía para el nuevo grupo creado.

O por poner otro ejemplo, dadas las preferencias de un turista, es posible que en un momento determinado cambie de grupo de turistas (y de guía asignado) para poder ir a puntos del mundo más acordes con sus gustos (exposiciones sobre un estilo concreto, visitas a monumentos más cortas, etc.).

---

### 6.3 *TOURISTORG*

---

TouristOrg es un sistema multi-agente basado en organizaciones. Para el diseño del sistema se han seguido los trabajos de [Agüero et al., 2009a] [Agüero et al., 2009b] [Agüero et al., 2010] en los que se aplica el enfoque MDD (*Model Driven Development*) en metodologías orientadas a organizaciones. Es posible diseñar la organización de forma unificada, intuitiva y con un alto nivel de abstracción. Mediante esta aproximación se facilita y se simplifica el proceso de diseño de SMA orientado a organizaciones obteniendo modelos de organizaciones virtuales que pueden ser implementados en diferentes plataformas. La idea fundamental es la de crear modelos diferentes para distintos niveles de abstracción, combinándolos para llegar a una implementación. Se ha utilizado GORMAS [Argente, 2008] como metodología de diseño y THOMAS [Carrascosa et al., 2009] [Giret et al., 2009] como plataforma final de desarrollo. En el Anexo A se encuentra la documentación completa de análisis, diseño de la estructura organizativa y diseño de la dinámica de la organización. Con estas fases se especifica cuáles son los servicios que ofrece la organización, cuál es su estructura interna y qué normas rigen su comportamiento. Siguiendo las pautas indicadas en la guía metodológica [Argente, 2008], una de las primeras tareas es instanciar la vista funcional (misión) del modelo de organización la cual representa los productos y servicios que ofrece el sistema, el tipo de entorno, los objetivos globales que persigue (misión y justificación), los grupos de interés a los que afecta y la información que consume. Puede observarse en la Figura 14.

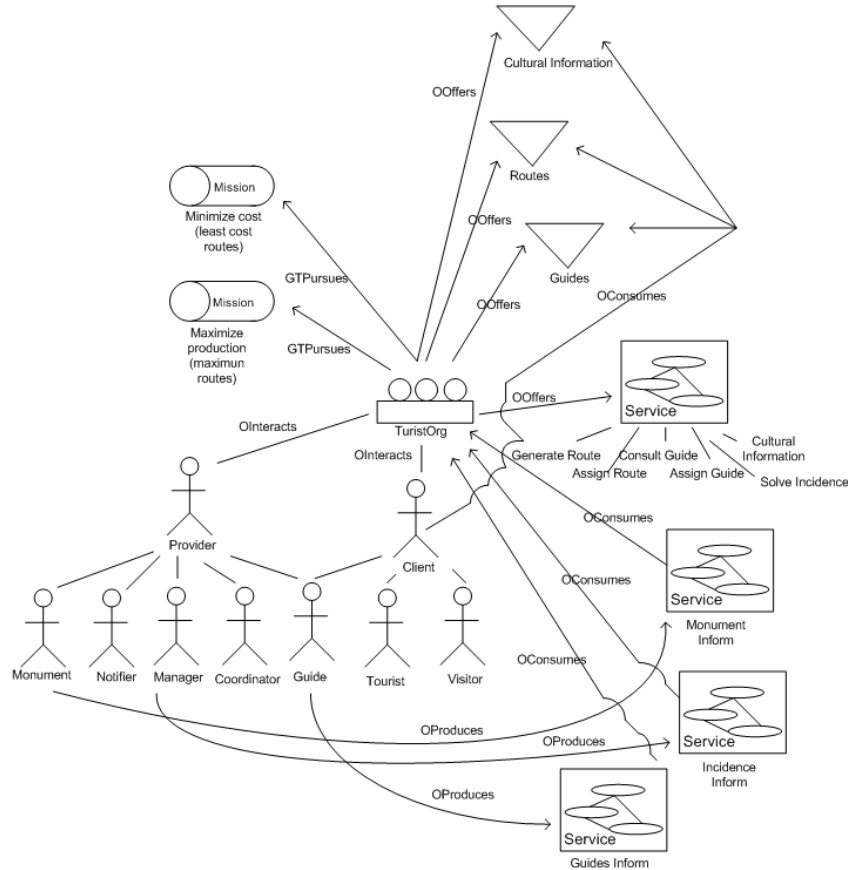


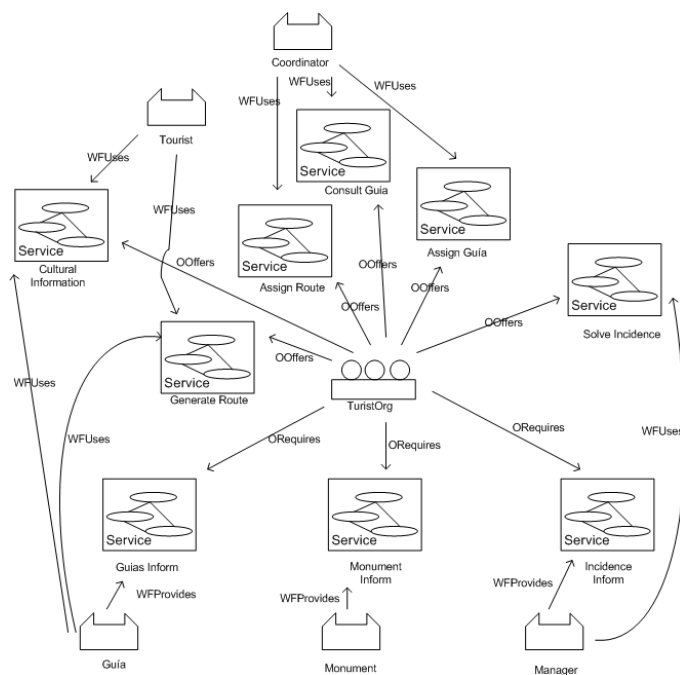
Figura 14. Diagrama del modelo de organización (vista funcional). Misión del sistema.

El sistema ofrece como producto principal las rutas turísticas, que serán requeridas por los guías o los propios turistas. También ofrece como productos la información cultural sobre los monumentos del sistema. El servicio encargado de informar sobre las características de cada monumento será tarea del agente monumento. Las guías también se ofrecen como "producto" ya que contendrán cierta información personal, es decir su perfil, generada por ellos pero que luego podrá ser utilizada y consultada por otros miembros de la organización para realizar asignaciones. En este caso, un guía podrá: (i) ser cliente cuando quiera obtener del sistema información sobre las rutas a seguir o sobre los monumentos que se les ha asignado; (ii) ser proveedor cuando ofrezca los planes de las rutas que ha generado en el sistema o cuando de información sobre su perfil "Servicio Información Guías". El sistema ofrecerá los servicios necesarios para generar y asignar rutas a los guías, consultar la información de los guías (perfil), asignar guías a turistas y consultar información sobre monumentos (información cultural). También podrá resolver incidentes (como por ejemplo que un monumento esté o no disponible



para generar rutas nuevas). Estos incidentes vendrán proporcionados por el Manager, que será el encargado de informar de todo lo que ocurre en y para el sistema. La misión de la organización será, por un lado minimizar el coste del sistema de guiado, intentando generar mejores rutas con menos recursos (tiempo , dinero, etc.); y por otro lado, maximizar la producción, es decir, generar más beneficios haciendo cuantas más rutas mejor. Los dos objetivos son complementarios y proporcionales, es decir, a mínimo coste con más rutas implica mayor producción.

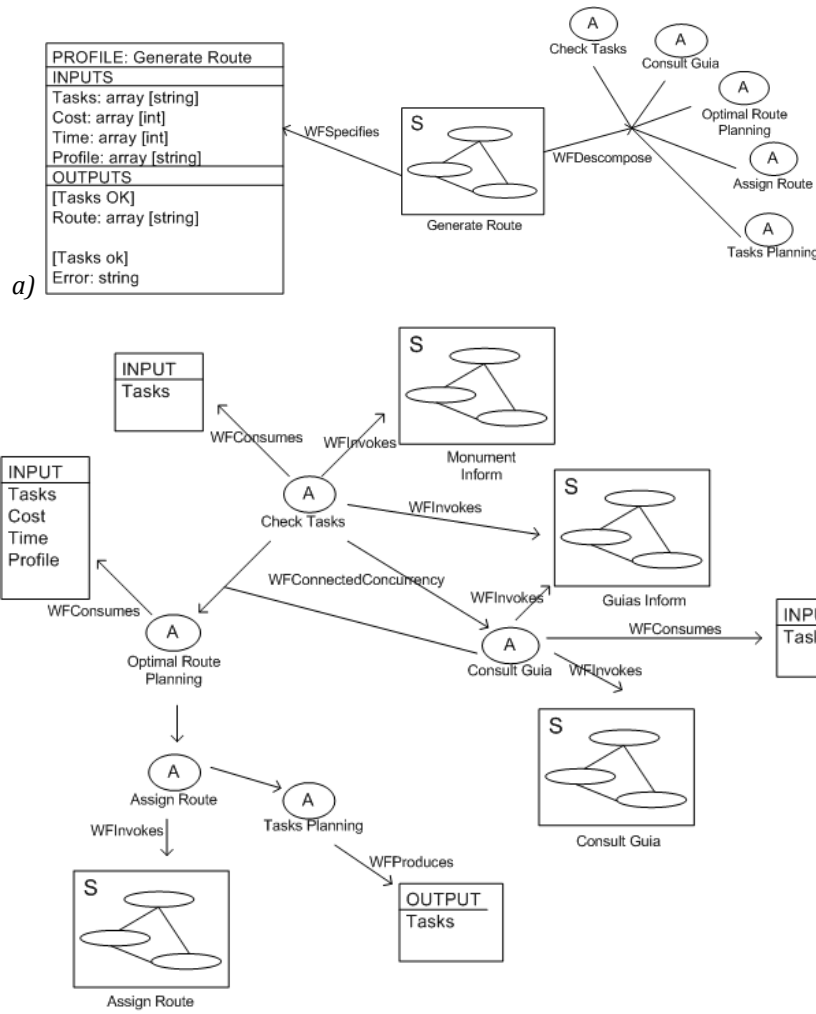
Siguiendo todas las pautas, obtenemos el modelado de la organización, desde su estructura interna hasta su funcionalidad. En el diagrama de la Figura 15 se muestra la vista funcional (funcionalidad externa) del modelo de organización asociado a TouristOrg, en el que todos los servicios conectados entre sí tienen asociados los roles del sistema con las relaciones WFUtiliza y WFProporciona. También requerirá de ciertos servicios que le proporcionen información. En este caso se representa la información que puede ser obtenida mediante la invocación a servicios proporcionados por agentes que no tienen por qué ser de la organización pero que pueden tomar los roles de Monumento, Guía y Manager para proveer esos datos a los agentes de la organización.



**Figura 15. Diagrama del modelo de organización, vista funcional, (funcionalidad externa)**

Se detallan cada uno de los servicios identificados en la vista funcional anterior y se crea el correspondiente modelo de actividad. Además se descomponen en

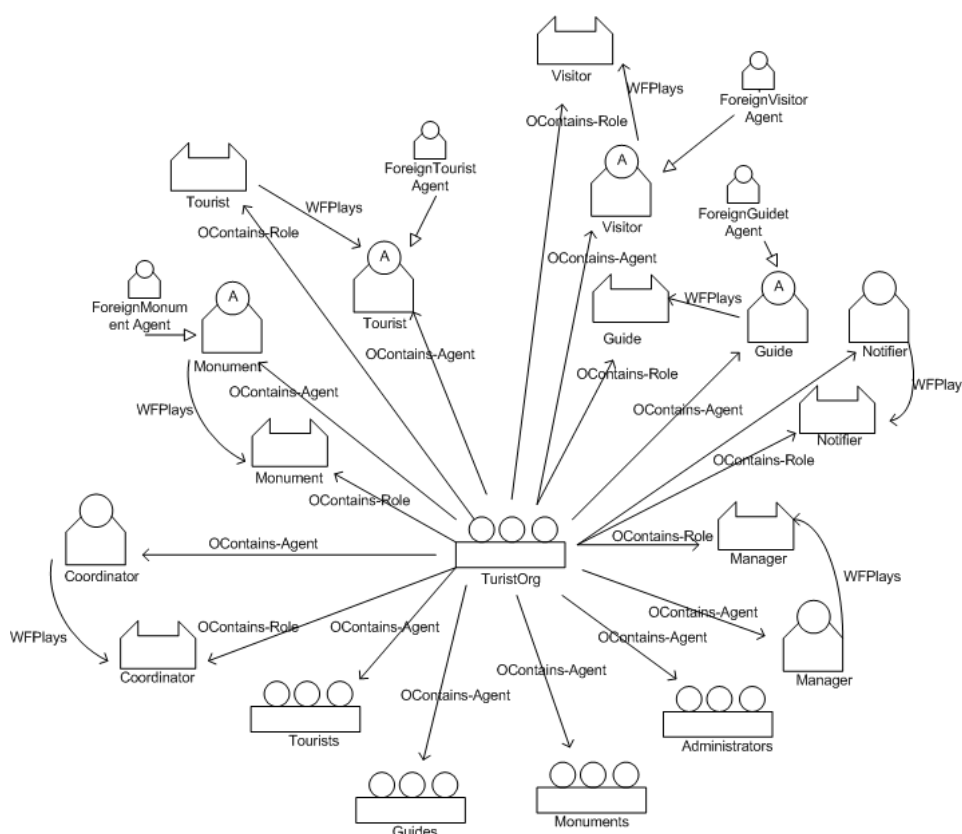
una serie de tareas que precisan para su funcionamiento (entidad A-Tarea y relación WFDescompone). La siguiente figura muestra el diagrama del modelo de actividad asociado al servicio Generar Ruta y las A-Tareas en las que se descompone.



**Figura 16. Diagrama del modelo de actividad del servicio Asignar Guía. a) Perfil del servicio b) Relaciones entre las A-Tareas que lo componen**

También se establece la funcionalidad ofrecida como sistema abierto, que incluye tanto los servicios que se deben publicitar como las políticas de adquisición y liberación de roles. Se determina qué funcionalidad debe ser implementada por agentes internos al sistema y cuál se publicita para poder ser suministrada por agentes externos. De este modo, se modela la dinamicidad de

la organización como sistema abierto. Los roles monumento, turista, visitante y guía son accesibles por agentes externos, requiriéndose un proceso de adquisición del rol. Los roles manager, notificador y coordinador se asocian a agentes internos, siendo no accesibles. En la Figura 17 se muestra el modelo de organización, indicando los agentes externos (A-Agentes), los internos (Agentes) y los roles que juegan.



**Figura 17. Diagrama actualizado del modelo de organización, en el que se muestran los agentes internos, externos y los roles que juegan.**

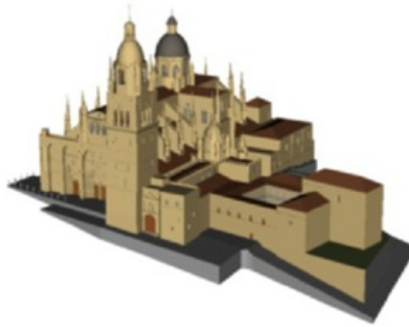
En nuestro caso, los roles de gestión y los servicios asociados a las unidades organizativas estarán disponibles gracias a la plataforma THOMAS [Carrascosa et al., 2009] [Giret et al., 2009]. El OMS (Organization Management Systems) ofrece los servicios necesarios para el adecuado funcionamiento de una organización. Facilita una serie de servicios para registrar o desregistrar componentes estructurales, en concreto, los roles, las normas y unidades existentes en el sistema. También ofrece servicios para informar sobre cuáles son dichos componentes.

---

## Evaluación y conclusiones

---

El entorno virtual en el cual habitarán nuestros agentes dispondrá de los inventarios de un conjunto documental de información turística basada en imágenes, locuciones, y demás contenidos. Para la simulación se han seleccionado 20 agentes virtuales representando visitantes, una media de 7 guías, y varios monumentos dinámicos y estáticos para nuestro enclave. Para el módulo de simulación se ha utilizado REPAST (*Recursive Porous Agent Simulation Toolkit*) [North et al., 2007][REPAST2009], un conjunto de librerías Java que permiten a los desarrolladores crear entornos de simulación, crear agentes en redes sociales, recopilar datos de las simulaciones automáticamente, y construir interfaces de usuarios fácilmente. Una descripción detallada de esta herramienta (y del resto de herramientas y entornos utilizados) puede verse en el Anexo B. La Figura 18 muestra 3 de los modelos tridimensionales utilizados para representar en el entorno virtual a tres de los agentes monumento identificados en el caso de estudio.





**Figura 18. Modelos tridimensionales utilizados para representar físicamente en el entorno agentes del tipo monumento (Catedral de Salamanca, Iglesia de San Esteban, Casa Las Conchas).**

Hay que resaltar que las simulaciones han sido realizadas con agentes Visitantes (no con Turistas) ya que no varían sus datos a lo largo del tiempo de ejecución de las pruebas. Para evaluar la adaptación del modelo en la replanificación y reorganización se han aprovechado las variaciones de los agentes Guías.

Los agentes que jueguen el rol de Guía serán los que obtengan dinámicamente la planificación, dependiendo de las tareas que necesiten llevar a cabo para cada grupo de visitantes. Los planes generados deben garantizar que todos los visitantes asignados a un guía realicen su ruta turística. Serán personalizados, dependiendo del perfil del Guía y los hábitos de trabajo, y deberán tener en cuenta restricciones relacionadas directamente con los agentes de manera individual, y restricciones de la organización. Estas últimas vendrán dadas por las normas propias de la sociedad de agentes. Veámos cómo, según el modelo planteado, en un momento concreto  $t$  de la vida de la organización pueden pasar tres tipos de situaciones que nos hacen llevar a cabo una replanificación de manera dinámica: puede suceder que un agente tenga un problema, o que cambie la estructura, o que se modifiquen las normas de la sociedad. Normas de la sociedad en nuestro caso de estudio son por ejemplo:

- ❖ el tiempo de trabajo de un agente guía (8 horas),
- ❖ el número de agentes turistas máximos asignados a un guía,
- ❖ el día de visitas para ciertos monumentos así como su horario de visita,
- ❖ el número de agentes guías máximo que pueden participar en una ruta,
- ❖ el número mínimo de puntos a visitar en una ruta.

Ante posibles cambios en las ejecuciones de las agentes, la sociedad debe ser capaz de reorganizarse dinámicamente, de adaptarse a ellos y continuar con planes que le permitan llegar a los objetivos establecidos en un inicio. Se

mencionaba anteriormente un ejemplo en el que era obligatorio formar grupos de turistas de 5 personas para entrar en cierto museo, y en un inicio los grupos estaban formados por 10 personas. En este supuesto el sistema debe ser capaz de reasignar los grupos y además introducir un nuevo agente que sea capaz de llevar a cabo el rol de Guía para el nuevo grupo creado. En este caso se produciría una reasignación global. Un ejemplo de replanificación individual dentro del caso de estudio es cuando al visitar un museo, se produce un retraso (en una situación real podría haber cola, obras en el museo, o algo similar). En este caso, el agente Guía que está realizando esa tarea puede replanificar su plan individual y dejar esa tarea, esa visita a ese museo, para otro momento. Se llevaría a cabo entonces una reordenación de sus tareas. En el caso de que no fuese posible sería necesario una reorganización con el resto de agentes Guía, se precisaría una reasignación global.

En la adaptación del sistema a los cambios, teniendo en cuenta el modelo de planificación, en todo momento se pretende minimizar un coste en la función objetivo; coste que será proporcional al tiempo trabajado de una agente Guía, a los recursos empleados y al perfil asociado. El perfil de un agente Guía está relacionado con su especialización. Sobre el agente será conocido:

- ❖ el tipo de monumento en el que está especializado (pintura, escultura, etc.),
- ❖ el idioma/as,
- ❖ el tamaño de grupo que puede tener a su cargo y
- ❖ el tipo de grupo en el que está especializado (niños, ancianos, etc.).

Se han tenido en cuenta estas opciones para el caso de estudio, pero podrían añadirse más según el nivel de detalle que quisiéramos darle al agente Guía. Así por ejemplo, podríamos tener que para un Guía especializado en un cierto tipo de monumento asignado en una ruta que no es su especialidad, se le asignaría un coste mayor que a otro agente Guía sí especializado en esa ruta.

Con respecto a la recuperación de fallos del sistema, indicar que existe una monitorización continua. Cada agente guarda sus datos dentro de una base de datos. Los agentes más sensibles al fallo son los agentes Turistas y los agentes Guías, así que guardan su estado cada hora. Cuando un agente falla, se genera en THOMAS otra instancia del mismo que se crea con la última copia de seguridad. La base de datos y el servidor deben tener capacidad de recuperación de fallos y control de redundancia así que para el servidor utilizado en el sistema se utiliza un sistema RAID (Redundant Array of Inexpresive Disk). Es en el servidor donde se ejecutará también la simulación del mundo virtual, actuando en colaboración con la arquitectura THOMAS (Figura 19).

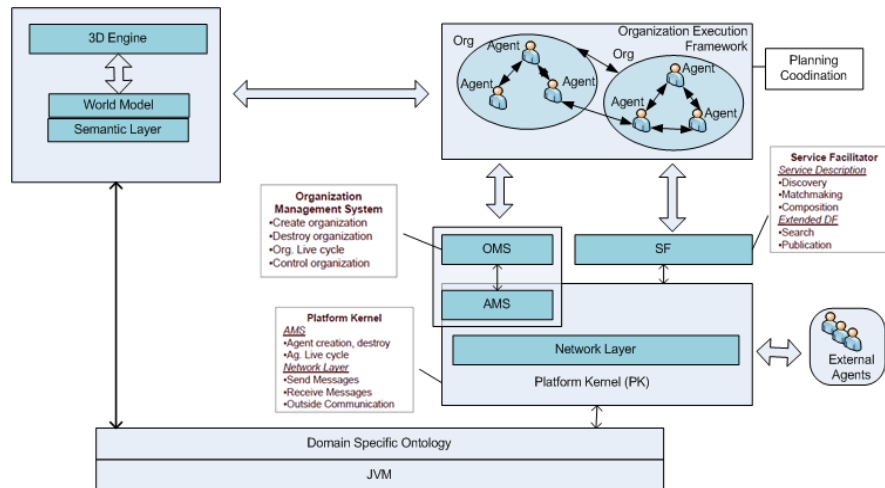


Figura 19. Módulos THOMAS

Como se ha indicado antes, el Anexo A proporciona la documentación completa de análisis y diseño del caso de estudio, y, en el Anexo B puede encontrarse una descripción de todas las herramientas utilizadas para los experimentos.

## 6.4 EJEMPLO DE PLANIFICACIÓN

Los agentes Guía son considerados los agentes planificadores de la organización. Los objetivos del agente Guía son, por un lado generar dinámicamente los planes que tiene que llevar a cabo, y por otro, mantener los informes de trabajo sobre las tareas que realizan. Todo esto para garantizar que los Visitantes asignados a los Guías realizan su ruta. Es decir, el agente Guía planifica el día de trabajo obteniendo planes dinámicamente dependientes de las tareas necesarias que soliciten los Visitantes. El agente Guía ofrecerá dentro de la arquitectura 3 servicios básicos:

- ❖ *ConsultarGuía*, para la consulta de los datos relativos a un Guía concreto.
- ❖ *ResolverIncidencia*, para solucionar incidentes (como por ejemplo que un monumento esté cerrado, o que un turista decida abandonar su ruta) y replanificar en el caso de que sea necesario.
- ❖ *GenerarRuta*, para la generación de la solución (plan) más replanificable para la organización de agentes dado un conjunto de preferencias para un problema.

Si tenemos un conjunto de creencias  $B$  compatible con un problema de planificación  $E$  que se desea llevar a cabo con la organización de agentes, es

posible generar un plan que contenga todos los posibles planes producidos por la combinación de las creencias compatibles.

Una tarea en un plan, no es más que un objeto java que contiene los datos de un agente Visitante o Turista que pide un servicio, la descripción del servicio y los límites de tiempo para llevarla a cabo. Por cada tarea, se establecen una o más metas. Una meta es también un objeto java, que identifica lo que un agente Guía quiere conseguir (tarea completada) y bajo qué restricciones. Para alcanzar sus metas se van generando planes, que son también procesos escritos en código java. Una meta puede crear nuevas metas (sub-metas). Por ejemplo, para la tarea "Visitar Museo 1", el agente Guía crea una nueva meta para cada ejecución concreta. El sistema CBP construye planes como secuencia de tareas que necesitan llevar a cabo los Guías. Una descripción del problema estará formado por las tareas que cada agente Guía necesita ejecutar, los recursos disponibles, y el tiempo asignado. Un ejemplo de tarea puede verse en la Tabla 3.

Tabla 3. Ejemplo de tarea

Task	Data
TaskId	66
TaskType	23
TaskDescript	Visit cathedral
TaskPriority	3
TaskGoal	0
TaskIncidents	0
TouristId	6
GroupTouristId	2
TouristIdiom	Spanish
MinTime	13:00
MaxTime	13:45
TaskResources	Tickets cathedral

El agente Coordinador de la organización es el encargado de asignar, para cada tarea  $j$ , el agente Guía  $i$  que la llevará a cabo, teniendo en cuenta para ello la base de casos en la que cada agente Guía almacena el coste que le supuso realizar esa tarea en el pasado. Una vez que el Coordinador identifica todos los agentes de la organización que son necesarios para llevar a cabo el plan con todas las tareas, asigna a cada uno de ellos las que debe realizar. En este momento, cada agente Guía conoce sus tareas a realizar y para ello diseña su plan individual. Cada agente Guía es un agente de tipo CBP-BDI que dispone de la capacidad de proveer en tiempo de ejecución planes eficientes. Para ello incorpora un planificador basado en casos (CBP). El propósito de estos agentes es adaptar soluciones que ya han sido resueltas en el pasado para resolver problemas nuevos. En este caso, los nuevos problemas estarán definidos por el conjunto de tareas que el agente Coordinador le ha asignado. Sabiendo las tareas que tiene que hacer, el agente Guía llevará a cabo su ciclo para generar el plan más replanificable en tiempo de ejecución (MRPI) [Corchado et al., 2008].



Si en un momento determinado de la ejecución uno de los agentes Guía no puede seguir con el plan que ha creado, intentará llevar a cabo una replanificación de sus tareas. Esto es, intentará generar otro plan con las tareas que el agente Coordinador le asignó. Si no es capaz de obtener un plan nuevo, tendrá que informar al agente Coordinador para que se lleve a cabo una reasignación de tareas entre el resto de compañeros de organización, es decir, entre el resto de agentes Guía. En este supuesto, el agente Coordinador tendrá en cuenta las tareas que se han llevado a cabo hasta ese instante y generará un nuevo plan global en el que cada agente Guía será reasignado con alguna de las tareas que queden por hacer. Puede verse un esquema de este funcionamiento en la Figura 12 del anterior capítulo. La organización es capaz de adaptarse con los mínimos cambios para su estructura. Intentará llevar a cabo una replanificación de tareas a nivel de agente y si esto no es factible, llevará a cabo una reorganización en la que se producirá una reasignación de tareas entre los agentes de la sociedad. Como último caso llevará a cabo la modificación de su estructura (por ejemplo, la inclusión de nuevos agentes o la reorganización de grupos de agentes). A continuación vamos a ver este proceso con más detalle.

Centrándonos en la generación de planes de cada agente Guía, es importante recalcar que en el estado de recuperación del ciclo CBP del agente, se recuperan las descripciones más parecidas. Para hacer esto, el agente Guía podrá aplicar varios algoritmos de similitud (coseno, clustering, etc.). En este paso, se recuperan de la base de creencias aquellas descripciones de problemas recuperadas dentro de un rango de similitud cercano al problema original. En nuestro caso se permite un 20% de tolerancia. Una vez la mayoría de las descripciones han sido recuperadas, se obtienen las soluciones asociadas a ellas. Una solución contiene todos los planes (secuencias de tareas) que se han llevado a cabo para alcanzar los objetivos del Guía para una descripción de problema (asumiendo que la replanificación es posible) en el pasado, así como la eficiencia de esa solución.

El agente Guía tiene el control de la ejecución del plan. Tiene en cuenta los objetivos y recursos necesarios para cada tarea, así como los objetivos que necesita realizar y los recursos disponibles para llevar a cabo su plan. Los objetivos o planes que cada Guía se hacen teniendo en cuenta las restricciones vistas anteriormente como el tiempo de trabajo (8 horas) o el tiempo disponible para un problema (tiempo para una visita turística). El tiempo disponible es una restricción del problema que también influirá en el hiperplano de restricciones, en concreto, en el rango de valores positivos del eje z. Recursos necesarios para las tareas son por ejemplo el dinero, los tickets de entrada a los museos o el equipo necesario para la visita a algún lugar turístico. Con esto, el agente Guía es capaz de ocuparse de los incidentes y de las interrupciones que puedan llegar a ocurrir durante la planificación.

Para cada agente Guía de la organización es necesario llevar a cabo la asignación de tareas. Es decir, los agentes Guía calculan sus planes óptimos

individuales sabiendo las tareas que tienen que llevar a cabo, pero en primer lugar cada uno de ellos tiene que saber qué tareas debe realizar. Es necesario realizar una asignación global en la organización de agentes en la que se distribuyan de una manera óptima las tareas entre todos los Guías de la organización. Esto será tarea del agente Coordinador. Para ilustrar cómo trabaja el agente Coordinador, vamos a ver un ejemplo.

En primer lugar es necesario tener en cuenta que cada agente Guía tiene diferentes perfiles de acuerdo a su cualificación y a las tareas que normalmente suele llevar a cabo. Fijando un estado “j”, para cada agente “i” con  $i \in \{1, \dots, m\}_{m \in \mathbb{N}}$ , se define la  $Z_{ij}$ -tupla, donde cada coordenada de la tupla se refiere a una característica que lo define, es decir, se considera el vector que almacena el perfil de un agente Guía en el mundo virtual. Se considera apropiado manejar los perfiles de los Guías ya que lo que se pretende es maximizar la eficiencia y para ello hay que minimizar los costes, y éstos dependen del perfil del agente Guía. Cuando se necesita llevar a cabo una asignación de tareas a los agentes Guías, se recuperan tanto las experiencias pasadas como el perfil del Guía, y se tienen en cuenta las necesidades del nuevo problema. Las tareas corresponden a un número de Turistas y/o Visitantes (un grupo) que serán asignados a uno de los Guías de la organización. En el perfil de un Guía se tiene en cuenta las especializaciones de ese agente. Por ejemplo, no todos los Guías son capaces de hablar todos los idiomas. Si un Guía está más cualificado en un idioma concreto será asignado al grupo que hable esa lengua, pero siempre teniendo en cuenta las restricciones como el tiempo de trabajo (8 horas); por lo que el número de turistas y visitantes asignados a un Guía dependerá del tiempo necesario para llevar a cabo la ruta turística. El agente Manager tendrá en cuenta el grado de satisfacción de los turistas al finalizar una ruta y la llegada de nuevos turistas a la organización.

En segundo lugar, es necesario almacenar dentro de la memoria de creencias el tiempo utilizado para cada tarea, descrito como  $t_{ij}$  tiempo que tarda el agente Guía “i” en completar el estado o tarea “j”, siendo  $t_{ij} = \text{Máx}_k \{t_{ijk}\}$  donde  $t_{ijk}$  indica el tiempo que tarda el agente “i”, en realizar la tarea “j” para el turista “k”. Tomando el máximo sobre “k” (tipo de turista), nos aseguramos que al guía le de tiempo a llevar a cabo la tarea que tiene que hacer con independencia del tipo de turista. Dichos tiempos, inicialmente se estiman. Para la estimación, se hacen simulaciones de visitas, antes de dar servicio a turistas.

De entre todos los Guías disponibles, se prioriza la selección del Guía con el perfil más adecuado (ya que solo es posible asignar cada tipo de tarea a los agentes Guías que son capaces de llevar a cabo ese tipo de tarea concreto). Una vez que la asignación de tareas a los agentes Guía ha sido completada, la asignación se comunica a cada uno de ellos. A partir de este momento,

comienza el proceso de planificación. El agente Guía debe tener en cuenta el tiempo del que dispone y el tiempo requerido para cada tarea que le ha sido asignada. Además, tiene en cuenta los recursos disponibles y necesarios. Con el objetivo de crear un nuevo plan, el agente Guía recupera de la memoria de creencias los casos que contienen una descripción de problema similar a la del problema actual y se combinan las soluciones (planes) que se utilizaron para resolverlos. Una gran cantidad de mediciones han de ser adoptadas con el fin de normalizar el tiempo necesario para llegar a un museo dado, o para llevar a un grupo de turistas y visitantes de un museo a otro. Estos tiempos están incluidos directamente en el tiempo asignado para cada tarea.

Un plan puede ser interrumpido por razones muy diferentes. Las que se han tenido en cuenta en el mundo virtual son: no se dispone de un recurso (no hay dinero para comprar tickets), un Guía no está disponible y hay que asignar otro (reasignación global), no es posible realizar una visita en el tiempo especificado (replanificación individual), una situación de emergencia, etc. Si un agente Guía se encuentra en la situación de que un plan es interrumpido, rechaza el plan inicial y propone un plan alternativo. La primera acción que intentará el agente es la replanificación de las tareas intentando mantener la asignación que originariamente hizo el agente Coordinador. El nuevo plan debe cumplir con los objetivos y restricciones iniciales. Si esto es imposible, será necesario realizar una reasignación de las tareas a los Guías. Esta reasignación se realiza minimizando los cambios y es necesario tener en cuenta las tareas que se asignaron a los Guías inicialmente, el desarrollo de los planes (las tareas se han realizado y las que todavía hay que hacer) y los perfiles de los Guías (dando prioridad a cualificación para realizar las tareas que no pudieron ser completadas). Cuando se introduce una nueva tarea, se realiza una reasignación y el Guía al que se le asigne esta tarea deberá replanificar para poder incluirla ahora en su plan.

Por último, dependiendo de la eficiencia que obtenga el plan, será almacenado junto con su nivel de eficiencia en la base de creencias para que pueda ser utilizado en problemas similares en un futuro. De la misma manera, las tareas llevadas a cabo por cada agente Guía también se almacenarán para que puedan ser utilizadas por el agente Coordinador para llevar a cabo la siguiente asignación de tareas de la organización. En el párrafo siguiente se muestra un ejemplo en detalle.

Sea  $E_g = \{e_0^g, \dots, e_n^g\}$  las tareas llevadas a cabo con un grupo de turistas y visitantes "g", puestas en orden de prioridad. Tenemos el siguiente problema

$E = \bigcup_g E_g = \{e_0, \dots, e_n\}$ , donde E denota el conjunto completo de tareas que tienen que ser llevadas a cabo (por esta razón no tienen superíndice). Supongamos que tenemos 10 guías. Seleccionando un Guía  $i \in \{1, \dots, 10\}$  al azar (en particular,  $i=3$ ), la asignación de tareas de acuerdo a su perfil es:

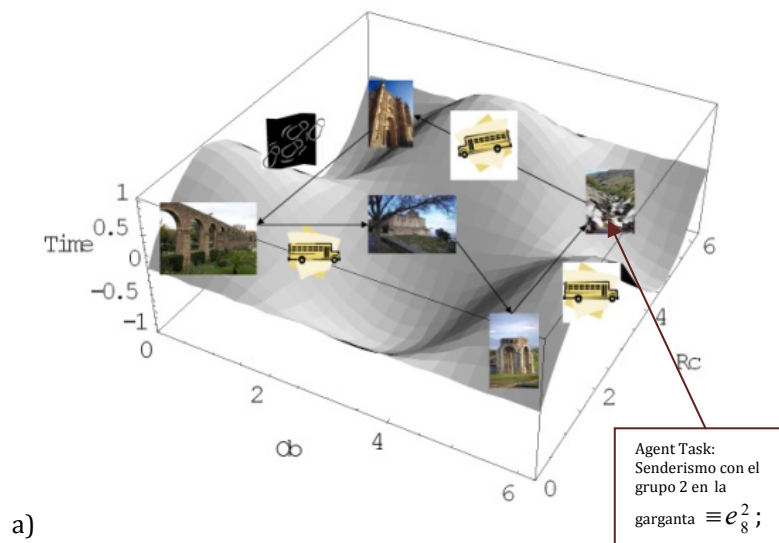
1. Agent Task: Visita con el grupo de turistas 2 en la catedral  $\equiv e_1^2; t_{31} = 30$  min.
2. Agent Task: Llevar al grupo de turistas 2 al acueducto  $\equiv e_2^2; t_{32} = 15$  min.
3. Agent Task: Llevar al grupo de turistas 2 a la ermita  $\equiv e_3^2; t_{33} = 10$  min.
4. Agent Task: Visita a la ermita  $\equiv e_4^2; t_{34} = 10$  min.
5. Agent Task: Llevar al grupo de turistas 2 a la ciudad romana  $\equiv e_5^2; t_{35} = 20$  min.
6. Agent Task: Visita a la ciudad romana  $\equiv e_6^2; t_{36} = 30$  min.
7. Agent Task: Llevar al grupo de turistas 2 a la garganta  $\equiv e_7^2; t_{37} = 50$  min.
8. Agent Task: Senderismo con el grupo 2 en la garganta  $\equiv e_8^2; t_{38} = 20$  min.
9. Agent Task: Llevar al grupo de turistas 2 de vuelta a la catedral  $\equiv e_9^2; t_{39} = 10$  min.

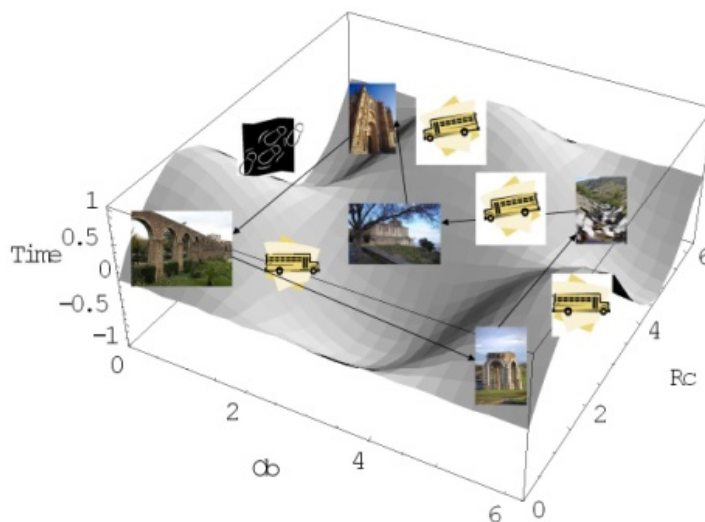
El cálculo de las tareas asignadas cumple que el tiempo total asignado al Guía no supera las 8 horas, así como el resto de restricciones que corresponden a las normas de la organización. Una vez la asignación de las tareas se completa, cada agente Guía lleva a cabo la generación y ejecución de su plan. Para ello, recupera asignaciones pasadas de características similares de la base de creencias, así como sus correspondientes planes que fueron utilizados para llevar a cabo esa asignación y se genera un nuevo plan. Para ejecutar el plan, en un entorno real, el agente Guía mostraría cada tarea en su dispositivo móvil. Una vez que una tarea es completada, el agente Guía solicita al Guía (humano) que introduzca el resultado obtenido después de que la tarea ha sido ejecutada. Cada tarea tiene una serie de objetivos que deben ser alcanzados para que el plan global sea completado con éxito. Para llevar a cabo cada tarea, el agente Guía debe tener un número de recursos disponibles. Por ejemplo, la tarea "Comprar tickets para el museo 1" corresponde con el objetivo "Visita al museo 1"  $\equiv O_0$  y "Desayunar, Comer, Tomar un refresco y Cenar" corresponde con el objetivo  $\equiv O_{2,4,6,7}$  (tarea 2 indica desayunar, tarea 4 comer, tarea 6, refresco y 7 cenar). La manera de codificar los recursos es similar. Se ha decidido que las variables de los objetivos y los recursos deberían ser dicotómicas (binarias) con valor de 0 o 1 para indicar la ausencia o presencia de un recurso o un objetivo y ser representada en la Figura 20a. El valor 1 indica que ese recurso es necesario o que este es el objetivo a ser alcanzado, mientras que 0 denota lo contrario. La Figura 20a muestra la representación del espacio en  $\mathcal{R}^3$  para las

tareas de acuerdo con las siguientes coordenadas: tiempo, número de objetivos conseguidos, y número de recursos usados. Específicamente, la Figura 20a muestra un hiperplano de restricciones y el plan seguido para un caso recuperado de la base de creencias, considerado similar. En ella, tenemos un grupo de turistas, que quieren hacer una visita al museo de la Catedral, al acueducto, a la ermita, al arco romano y alrededores y a una garganta. El desplazamiento entre la Catedral y el acueducto se puede hacer andando, pero el resto hay que hacerlo en autobús. Por las localizaciones geográficas de los puntos de interés, horarios de apertura y de cierre. Las restricciones son:

- ❖ La catedral tiene celebraciones litúrgicas, horario de entrada y salida del museo.
- ❖ La ermita no tiene museo, pero sí celebraciones litúrgicas y un bello paisaje a su alrededor.
- ❖ El acueducto, el arco y la garganta han de verse con luz solar.

Las posibilidades de rutas son muchas, hay 5! posibilidades es decir, 120 rutas posibles; las cuáles no son todas viables, debido a las restricciones anteriores.





b)

Figura 20. a) Representation of a space  $\mathcal{R}^3$  for tasks . b) Representation of a space  $\mathcal{R}^3$  for replanned tasks.

En una simulación en la que el Coordinador asignó este grupo de turistas a un Guía, la planificación del Guía para las tareas a desarrollar fue la que se muestra en la Figura 20a.

Se concretaba un tiempo con el grupo en la puerta de la Catedral, se hacía una visita guiada al museo de la Catedral (30 minutos), después se iba andando al acueducto (15 minutos); desde allí en autobús (10 minutos) se hacía una visita a una ermita (10 minutos), se volvían a montar los turistas en el autobús (20 minutos) y se dirigían hacia las ruinas de la ciudad romana (30 minutos de visita). Después se volvía a montar en autobús (50 minutos), para llegar y hacer un poco de senderismo en una garganta (20 minutos); último punto de la visita y regreso a la puerta de la catedral en autobús (30 minutos). Dicha ruta es la más replanificable, por la localización de la ermita. Cuando se iba a coger el autobús en el acueducto, se nos informa de que en la ermita hay una boda y que la novia ha llegado con retraso; así que para no coincidir con la celebración, se replanifica la ruta a seguir, siendo el plan resultante se muestra en la figura 6b.

Se dirige a los turistas en autobús (15 minutos) hacia las ruinas de la ciudad romana (30 minutos de visita). Después se vuelve a montar en autobús (50 minutos), para llegar y hacer un poco de senderismo a una garganta (20 minutos); desde allí en autobús (20 minutos) se hace una visita a una ermita

(10 minutos), se vuelve a montar los turistas en el autobús (10 minutos) y último punto de la visita y regreso a la puerta de la catedral en autobús. La asignación de tareas para el Guía 3 quedaría entonces de la siguiente forma:

1. Agent Task: Visita con el grupo de turistas 2 en la catedral  $\equiv e_1^2$ ;  $t_{31} = 30$  min. Ya realizada.
2. Agent Task: Llevar al grupo de turistas 2 al acueducto  $\equiv e_2^2$ ;  $t_{32} = 15$  min. Ya realizada.
3. Agent Task: Llevar al grupo de turistas 2 a la ciudad romana  $\equiv e_5^2$ ;  $t_{35} = 15$  min.
4. Agent Task: Visita a la ciudad romana  $\equiv e_6^2$ ;  $t_{36} = 30$  min.
5. Agent Task: Llevar al grupo de turistas 2 a la garganta  $\equiv e_7^2$ ;  $t_{37} = 50$  min.
6. Agent Task: Senderismo con el grupo 2 en la garganta  $\equiv e_8^2$ ;  $t_{38} = 20$  min.
7. Agent Task: Llevar al grupo de turistas 2 a la ermita  $\equiv e_3^2$ ;  $t_{33} = 20$  min.
8. Agent Task: Visita a la ermita  $\equiv e_4^2$ ;  $t_{34} = 10$  min.
9. Agent Task: Llevar al grupo de turistas 2 de vuelta a la catedral  $\equiv e_9^2$ ;  $t_{39} = 10$  min.

Para que la Figura 20 no fuese demasiado grande, y para que el plan pudiese ser apreciado a primera vista, el eje de tiempo ha sido redimensionado (eje z), estableciendo un isomorfismo entre los intervalos  $[0, 1]$  y  $[0, 8]$ . El isomorfismo es el siguiente:

$$\begin{aligned} \lambda : [0, 1] &\rightarrow [0, 8] \\ z &\rightarrow \lambda(z) = 8 \cdot z \end{aligned} \quad (6.1)$$

Para los casos recuperados similares, el procedimiento es el mismo. El nuevo plan de un agente es generado en secciones de tal manera que tiene la mayor densidad de planes alrededor.

Las figuras muestran el plan llevado a cabo. Para comprender la representación gráfica, vamos a centrarnos en la tarea inicial  $e_1$  y en la final  $e_9$ . Entre estas dos tareas el agente Guía podría llevar a cabo otras tareas que envolverían los mismos o diferentes turistas y visitantes implicados.

La idea que el modelo de planificación presenta es elegir como solución óptima el plan que tiene más planes a su alrededor. Así, en la Figura 20b, el plan

elegido es el plan con más planes a su alrededor y que tienen las tareas que podrían ser asignadas en caso de interrupción.

Otro plan propuesto para nuestro ejemplo fue:

- ❖ Organization task: Asignar Guía 1 al grupo 2, Guía 2 al grupo 4, Guía 3 al grupo 5, Guía 4 al grupo 6.
- ❖ Agent Task: Llevar al grupo de turistas 2 a la catedral  $\equiv e_1^2$ ;  $t_{11}=30$  min.
- ❖ Agent Task: Visita a la catedral del grupo de turistas 2  $\equiv e_3^2$ ;  $t_{13}=60$  min.
- ❖ Agent Task: Almuerzo del grupo 2  $\equiv e_2^2$ ,  $t_{12}=60$  min.
- ❖ Agent Task: Almuerzo de los grupos 4, 5 y 6  $\equiv e_2^4, e_2^5, e_2^6$ ,  $t_{v5}=120$  min con  $v \in \{2, 3, 4\}$
- ❖ Agent Task: Paseo por e río de los grupos 2,4,5 y 6,  $\equiv e_4^2, e_4^4, e_4^5, e_4^6$ ,  $t_{s4}=120$  min con  $s \in \{1, 2, 3, 4\}$

Durante la visita a la catedral es necesario reorganizar la estructura del sistema ya que el límite de personas por grupo es de máximo cuatro personas. Un guía nuevo tiene que aparecer en el plan ya que el grupo 2 está formado por 6 personas, y además con la condición de que en su perfil incluya como propiedad conocer el idioma francés ya que todos los turistas del grupo 2 hablan esta lengua. El nuevo plan quedaría de la forma siguiente:

- ❖ Organization task: Asignar Guía 1 al grupo 2.
- ❖ Agent Task: Llevar al grupo de turistas 2 a la catedral  $\equiv e_1^2$ ;  $t_{11}=30$  min.
- ❖ Organization task: Nueva formación de Unidad de visitantes: grupo 7.
- ❖ Organization task: Asignar Guía 5 al grupo 7.
- ❖ Agent Task: Visita a la catedral del grupo de turistas 2 y 7  $\equiv e_3^2, e_3^7$ ;  $t_{b3}=60$  min con  $b \in \{1, 5\}$
- ❖ Agent Task: Almuerzo del grupo 2  $\equiv e_2^2$ ,  $t_{12}=60$  min.
- ❖ Agent Task: Almuerzo de los grupos 4, 5 y 6  $\equiv e_2^4, e_2^5, e_2^6$ ,  $t_{v5}=120$  min con  $v \in \{2, 3, 4\}$
- ❖ Agent Task: Paseo por e río de los grupos 2,4,5 y 6  $\equiv e_4^2, e_4^4, e_4^5, e_4^6$ ,  $t_{s4}=120$  min con  $s \in \{1, 2, 3, 4\}$



En este ejemplo, la organización es capaz de adaptarse en el tiempo a los cambios que influyen en sus agentes miembros. Los agentes Guía son capaces de realizar planificación de tareas, son también capaces de replanificar, y el agente Coordinador es capaz de reorganizar las tareas de nuevo entre todos ellos en el caso de que no sepan adaptarse al cambio.

Los cálculos matemáticos para obtener  $h(x)$ , se llevaron a cabo con técnicas de Duchon; el conjunto de geodésicas  $\{g_n(x)\}_{n \in N}$  a través de la resolución de las ecuaciones de Euler y ecuaciones de transversabilidad; y para obtener el campo de Jacobi, se utilizó Mathematica ® 5.1 y las bibliotecas Jspline + y Jlink para Java.

## 7 RESULTADOS EXPERIMENTALES, CONCLUSIONES Y TRABAJO FUTURO

---

En este trabajo de tesis se ha presentado un nuevo modelo de coordinación para organizaciones de agentes cuya principal novedad es su capacidad de planificación dinámica y adaptativa para llevar a cabo una distribución óptima de las tareas de los agentes miembros de la organización. Se trata de un modelo único en su concepción, que permite dotar a una organización de capacidades auto-adaptativas en tiempo de ejecución para entornos altamente dinámicos. En resumen, el modelo presentado permite definir las acciones que una organización de agentes debe llevar a cabo, considerando los cambios que pueden producirse durante la ejecución de un determinado plan. Es una planificación adaptativa dentro de una organización de agentes.

Las siguientes subsecciones muestran algunos de los resultados experimentales obtenidos, las conclusiones de la investigación, las contribuciones principales a la tecnología de agentes y SMA, y algunas líneas de trabajo futuro a corto, medio y largo plazo.

---

### 7.1 RESULTADOS

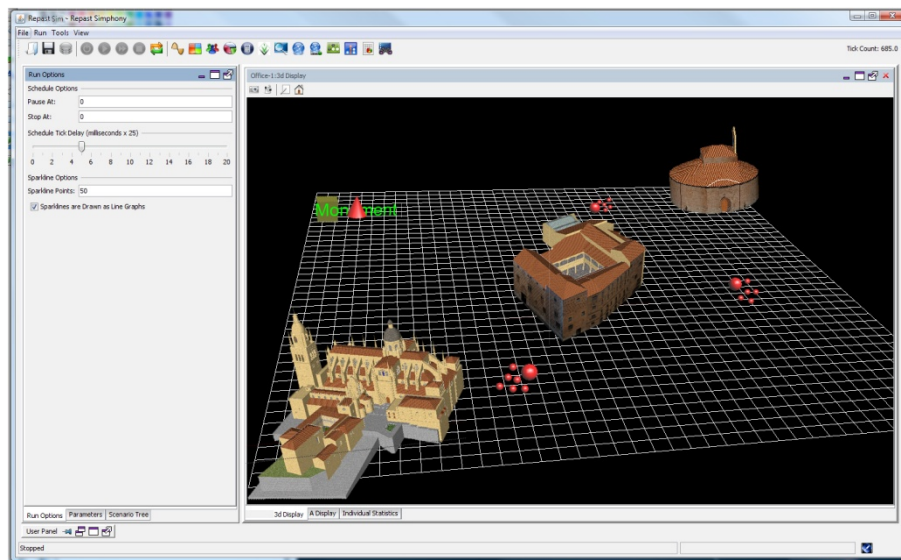
---

No cabe duda de que en los resultados de este trabajo tiene un peso importante el caso de estudio diseñado para la organización. Sin embargo, el objetivo del mismo no es la simulación precisa del comportamiento de un entorno turístico, sino la validación del modelo de planificación propuesto mediante la simulación de la organización del caso de estudio. Para poder interpretar los resultados convenientemente, hay que analizar con detenimiento el funcionamiento del escenario propuesto, con el fin de poder evaluar el impacto del nuevo mecanismo coordinador y planificador.

El caso de estudio se ha creado como un sistema de guiado turístico en el que se dispone de una media de 7 agentes guía, 6 enclaves turísticos, teniendo en cuenta los considerados dinámicos (conciertos, eventos, etc.) y los estáticos (iglesias, museos, etc.) y 20 agentes turistas/visitantes. Las principales pruebas han consistido en una serie de simulaciones ejecutando el mecanismo planificador de tareas. La memoria de casos ha sido depurada en cada simulación para eliminar la capacidad de aprendizaje del mecanismo, lo que fuerza al coordinador a realizar todo el proceso de asignación de tareas.

El módulo de simulación ha sido implementado en REPAST [North et al., 2007][REPAST2009], un conjunto de librerías Java que conforman una herramienta de modelado basado en agentes. La elección de REPAST se debe a

que es una herramienta de modelado ampliamente validada y utilizada por la comunidad científica que, a nuestro juicio, facilita la implementación y el análisis de cualquier modelo computacional basado en agentes. En el Anexo B se realiza la descripción detallada de esta herramienta, de su utilización en este trabajo, y un estudio comparativo de herramientas que proporcionan características similares. Tanto el módulo de simulación como el sistema de agentes han sido ejecutados sobre una máquina con las siguientes características: Sistema Operativo Windows Vista Business, SP2, x64, CPU 2,8 GHz, 6GB RAM. Todas las animaciones fueron renderizadas en tiempo real usando REPAST en un PC de estas características y con una tarjeta gráfica *Intel Series Express Chipset Family*. La siguiente figura muestra una captura de pantalla del módulo de simulación. En ella puede observarse tres de los modelos en tres dimensiones utilizados para la simulación del enclave, junto con varios modelos adicionales y tres grupos de turistas con su agente Guía correspondiente. Repast proporciona funcionalidades adicionales desde su entorno de ejecución, como por ejemplo la posibilidad de modificar las características de los agentes, cambiar propiedades, definir la velocidad de simulación (menú izquierdo que se muestra en la figura), visualizar diferentes vistas (3D, 2D, estadísticas en tiempo real) o la posibilidad de ejecutar herramientas adicionales como Matlab, VisAd, iReport o Weka. Esto abre un vasto campo de ampliación en este estudio, ofreciendo la posibilidad de creación de módulos adicionales de análisis de los sistemas simulados.



**Figura 21. Captura de pantalla del módulo de simulación**

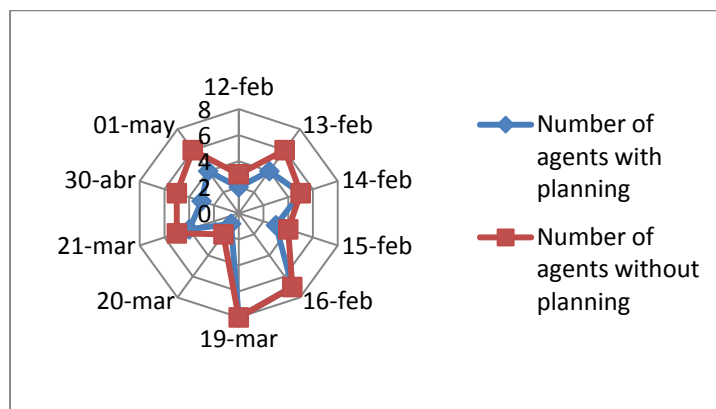
El sistema se ha probado en condiciones de simulación, con datos obtenidos a partir de distintas rutas aleatorias. El primer prototipo se implementó en Octubre de 2009. Los resultados obtenidos corresponden con los últimos

experimentos realizados. Dichos experimentos tenían una duración de entre media y 1 hora dependiendo del tiempo (y correspondientes ticks) que se querían simular. Por ejemplo, para el primer experimento mostrado, la simulación equivale a 2 meses y 18 días. La equivalencia de tiempo en REPAST puede ser ajustada dependiendo del número de *ticks* ("unidad de tiempo" en REPAST). Si por ejemplo consideramos 3600000 milisegundos por *tick*, significaría que un tick equivale a una hora en el calendario estandar (Gregoriano). En la simulación, un tick tarda un segundo real, por lo que la simulación de 2 meses y 18 días equivale a 31,2 minutos reales (dependiendo siempre de las condiciones de renderizado).

La principal ventaja de los modelos basados en agentes es que constituyen, en la práctica, un laboratorio computacional que nos permite obtener mediante simulación cuanta información y datos deseemos. Es responsabilidad nuestra, por tanto, hacer un adecuado diseño de experimentos que permita responder a las cuestiones de nuestra investigación. En nuestro caso, la validación del modelo de planificación.

La principal característica del modelo que se propone es que describe un proceso en el que el estado final que se alcanza depende de la secuencia de ocurrencias históricas (de naturaleza aleatoria) que suceden durante la simulación, aunque las condiciones iniciales puedan ser siempre iguales para todas las simulaciones. Esta naturaleza no determinista del sistema dificulta el análisis estadístico del mismo, aunque esto no significa que no podamos llegar a describir las propiedades de cualquier estado del sistema.

De cara a evaluar el mecanismo de coordinación propuesto, y teniendo como objetivo que se completen una serie de tareas consistentes en visitar los mismos puntos turísticos, en el mismo día; para el mismo número de turistas por grupos, en un grupo se usó el mecanismo de planificación y en otro no.



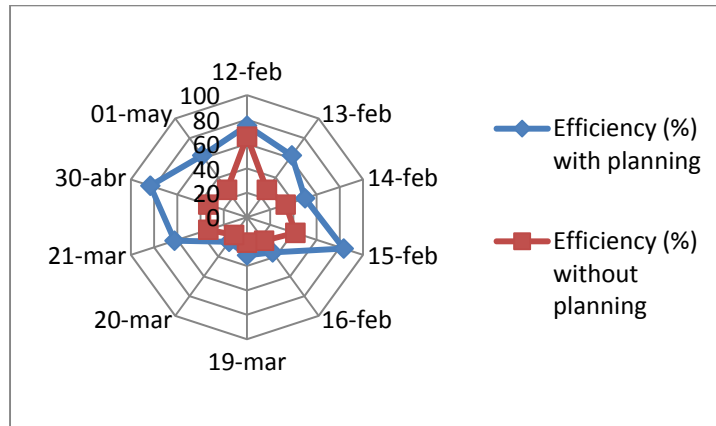


Figura 22. a) Número de agentes trabajando simultáneamente b) Eficiencia en la organización de agentes con y sin planificación

Los resultados para diferentes días, en cuanto a número de agentes guía empleados pueden observarse en la Figura 22a. La simulación equivale a 2 meses y 18 días de los que se representan los valores de eficiencia y número de agentes para 10 de las jornadas de trabajo. El color azul representa la media de agentes guía requeridos cada día usando el planificador, y en rojo, sin utilizarlo. El modelo de coordinación propuesto en este trabajo de tesis supone una mejora sustancial con respecto a los mecanismos manuales existentes, ya que ayuda a la organización en la mejora en la coordinación global, permitiendo una asignación óptima de las tareas a realizar. Este hecho permite que sólo los agentes necesarios sean empleados para la realización de determinadas tareas y que los agentes puedan minimizar el tiempo empleado en sus jornadas, y por tanto, los costes de la organización.

Un punto importante para evaluar la validez del planificador, es comparar la eficiencia de cuando se emplea a cuando no, de cara a evaluar el incremento de eficiencia que se puede conseguir al incorporar el mecanismo planificador en el sistema. Se recuerda que la eficiencia definida en este trabajo de tesis doctoral relaciona el coste de usar los agentes con el número de puntos visitados. Considerando el caso anterior, en el que era fijo el número de puntos a visitar; gráficamente se muestra para esos días la eficiencia de ambos casos (Figura 22b). Tal y como se puede apreciar en la Figura 22b, la eficiencia del sistema se reduce notablemente si no se utiliza el mecanismo planificador.

Siguiendo esta línea de experimentos, también se observó el tiempo usado por el mecanismo planificador para calcular las rutas asignadas a los agentes planificadores dependiendo del número de agentes que formaban el sistema. En este caso, las pruebas se llevaron a cabo utilizando 1, 3 o 7 agentes guías y 5, 10 y 20 turistas simultáneamente. El número de rutas planificadas eran las equivalentes a 2 meses de trabajo, por lo que se tenía 20 rutas/mes x 2 meses =

40 rutas de 10 tareas cada una de ellas (*40Routes, 400 Agent Tasksy Organization Tasks*). En el experimento pudo observarse cómo al aumentar el número de guías, el modelo propuesto en este trabajo de tesis tardaba un poco más en asignar las rutas y los grupos, pero a pesar de esto, era muchísimo más rápido que un sistema sin planificador a la hora de completar las rutas asignadas. En la Figura 23 pueden verse estos resultados. En el eje X está representado el número de turistas utilizados (5, 10 y 20) mientras que en el eje Y se representa el tiempo en unidades de simulación (Ticks). Cada línea representa la simulación realizada para el número de turistas que indica el eje X y para 1, 3 y 7 guías en cada simulación. Primero sin utilizar el mecanismo planificador y posteriormente utilizándolo.

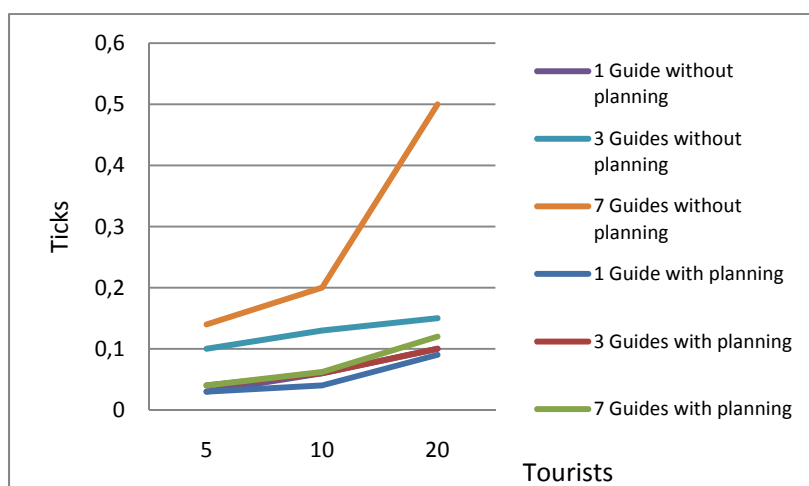


Figura 23. Tiempo (Ticks) vs. Número de agentes Guía y Turistas

Los resultados mostrados hasta ahora hacen hincapié en aspectos funcionales del sistema depurando la memoria de casos en cada simulación para eliminar la capacidad de aprendizaje del mecanismo, lo que fuerza al coordinador a realizar todo el proceso de asignación de tareas. Pero también es necesaria una evaluación del mecanismo desde el punto de vista del aprendizaje. En este sentido, los agentes CBR-BDI y CBP-BDI utilizan, a bajo nivel de implementación, un sistema de razonamiento basado en casos o de planificación basada en casos, por lo que el sistema desarrollado viene caracterizado por trabajar con casos. Para facilitar la construcción de agentes CBR-BDI y CBP-BDI en [Corchado y Laza, 2003] [Bajo et al., 2006] se presenta un conjunto de equivalencias entre las dos tecnologías a través de una biblioteca genérica. Un caso, en términos de CBR, consiste en un problema (situación inicial y un conjunto de metas) y la solución efectuada para solucionarlo. La definición de caso que se proporciona en la biblioteca CBR-BDI proporciona una gran potencia y flexibilidad al desarrollador, permitiéndole adaptar el caso a su problema concreto. Por ejemplo, para el problema de la

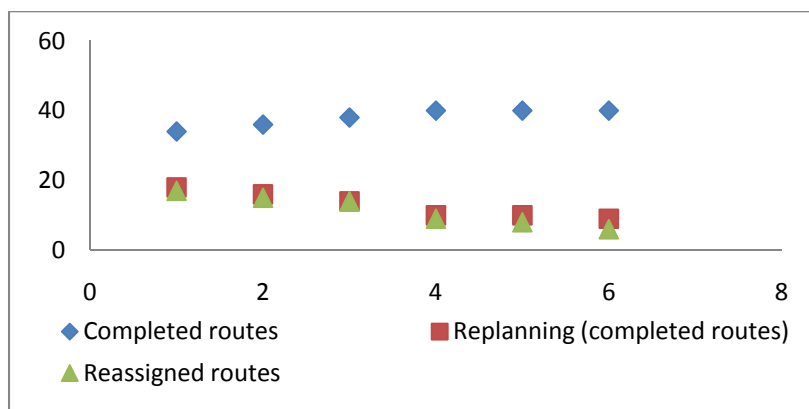
planificación de rutas turísticas, se define la descripción de un problema inicial para el que es necesario proponer una sugerencia de guiado a través de la clase *problemData* que constará de los siguientes atributos:

**Tabla 4. Atributos de los casos**

<b>Campo del Caso</b>	<b>Tipo de Elemento</b>
TOURIST	Perfil de turista (TouristProfile)
MONEY	Dinero disponible (Money)
TIME	Tiempo (Time)
INIT	Localización inicial del usuario (Location)
PREF	Preferencias del turista (Preference)

La Tabla 4 muestra una descripción del problema indicado mediante los atributos TOURIST, MONEY, TIME, INIT y PREF. Mediante estos atributos, el sistema es capaz de obtener la descripción de un problema que es necesario resolver. Además de estos atributos, la clase *problemData* proporciona los métodos que permiten actuar sobre el valor de dichos atributos. Los agentes CBR-BDI y CBP-BDI son agentes que trabajan con creencias, deseos e intenciones desde un punto de vista conceptual y casos desde un punto de vista de implementación. Un caso está formado por una descripción del problema inicial con atributos como los descritos en la Tabla 4, además de por los planes ejecutados para solucionarlo, las soluciones propuestas y la eficiencia para la solución final.

El modelo propuesto en el marco de esta investigación permite realizar replanificaciones y reorganizaciones dinámicas en tiempo de ejecución. Esto supone una gran ventaja en la elaboración de planes, ya que en el caso de que se produzca una interrupción no supone necesariamente desandar lo andado, sino que se obtienen alternativas que permitan continuar con el desarrollo del plan original. El proceso de planificación busca obtener una base de planes óptimos para la resolución de un problema. De entre estos planes se seleccionará aquel plan con una mayor densidad de planes a su alrededor. En el caso de que se produzca una interrupción, los agentes planificadores de la organización podrán recurrir a los planes cercanos para tratar de finalizar las tareas pendientes del plan original. En el caso de que esto no sea posible, se producirá una reasignación de tareas a los planificadores de la organización, para poder completar las tareas asignadas. CBP-BDI es una herramienta muy apreciada que optimiza la coordinación global del sistema. La siguiente figura muestra cómo el número de planes que se completan con éxito crece a medida que el sistema resuelve nuevos casos. Esto es debido a la capacidad del sistema para aprender y adaptarse a las experiencias adquiridas. De la misma forma, las experiencias obtenidas permiten una mejor adaptación al perfil del turista.



**Figura 24. Evolución del planificador**

En la Figura 23 es posible apreciar los resultados medios obtenidos para los agentes guía en el experimento. Para este experimento, cada simulación se hizo con el cálculo de 40 rutas correspondientes a 2 meses de trabajo, y todas con 7 agentes guía y 20 agentes turistas. En la Figura 23 se puede observar cómo el número de reasignaciones desciende a medida que el sistema adquiere experiencia (el eje X representa el número de simulaciones realizadas y el eje Y el número de rutas). El porcentaje de planes asignados no completados desciende desde un 19,1% a un 6,5%. Estos datos demuestran la gran capacidad de adaptación del sistema a medida que adquiere experiencia. Este tipo de comportamiento es habitual en los sistemas de razonamiento basado en casos, en los que generalmente es necesario un espacio de tiempo inicial para comenzar a adquirir la experiencia necesaria para ajustar su funcionamiento al entorno de trabajo.

Los resultados del estudio sitúan el modelo, junto con la arquitectura presentada en este trabajo de tesis (THOMAS) como una alternativa predominante potente con respecto al resto de arquitecturas para el desarrollo de organizaciones de agentes (Tabla 5), ya que se ofrece al menos las mismas capacidades que el resto de arquitecturas existentes y, en muchos casos, las mejora. Además, se ofrece la posibilidad de modelar comportamientos deliberativos y sociales y se incorpora un mecanismo propio de reorganización y adaptación social. THOMAS facilita el desarrollo de SMA en un paradigma organizacional y el modelo social añade las funcionalidades de reorganización y adaptación. Además, la diversidad de elementos influyentes en la generación de planes que presenta, lo hace más fácilmente adaptable a diversos contextos y problemas. Los dos siguientes experimentos proporcionan resultados comparativos entre el modelo y la plataforma utilizados y otros mecanismos existentes.

Como se indicó en la sección 3.2 (referente a coordinación), existen diferentes formas para llevar a cabo la asignación y reorganización de tareas. Parece



interesante comparar el modelo propuesto frente a otros tipos de planificación, abriendo así un camino de investigación futuro para la mejora del mismo. La asignación y reorganización de tareas en el mecanismo de planificación propuesto podrían haberse llevado a cabo de múltiples formas. Tres de estas formas pueden verse en la Figura 26.

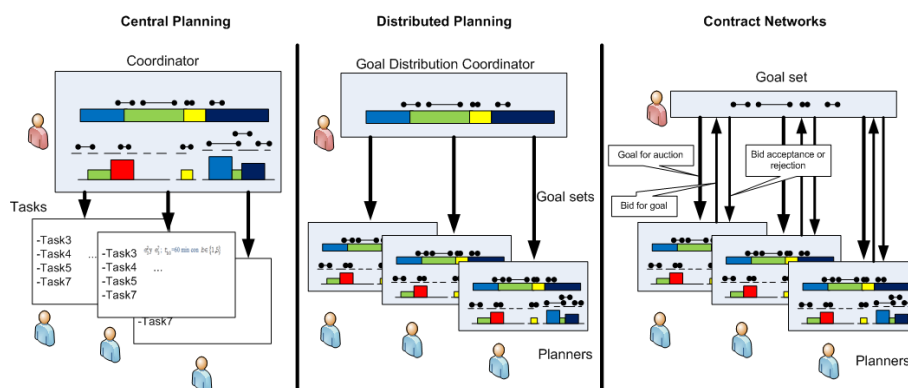


Figura 25. Diferentes enfoques para distribución de tareas.

Para la evaluación de resultados, se quiso comparar dos de estas posibilidades, para comprobar cómo se comportaría el sistema si en lugar de llevar a cabo una coordinación distribuida, se hubiese empleado un modelo en el que la planificación de todas las tareas de la organización era llevada a cabo de manera centralizada:

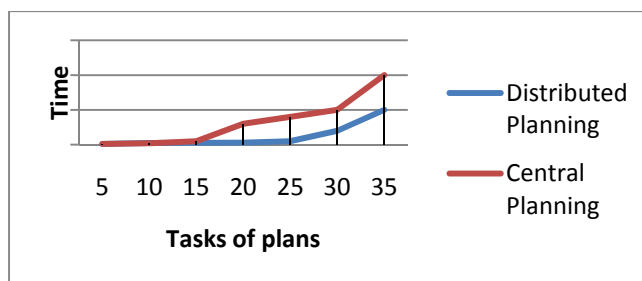


Figura 26. Planificación distribuida vs. Planificación central respecto al tiempo.

Utilizando una planificación centralizada para manejar la organización, se dispone de un agente líder (coordinador y planificador) encargado de generar los planes que serán posteriormente ejecutados por los demás agentes. Para asegurar que cada plan es correcto, la organización necesita adquirir grandes cantidades de información sobre los turistas, los guías y los monumentos para determinar si esos seguidores (guías) pueden llevar a cabo sus planes apropiadamente.

El esquema de planificación distribuida reduce la cantidad de información necesaria distribuyendo los objetivos. En este caso, el coordinador tendrá en cuenta un subconjunto de información sobre estados colectivos, con menos precisión de información sobre los agentes guías y produce un plan abstracto con suficiente detalle como para distribuir tareas objetivo entre los guías. A cada guía, le transmite su propio objetivo y éste a su vez, generará su solución óptima para llevarlo a cabo.

Otra forma de realizar la coordinación sería la utilización de redes de contratos en los que un líder anuncia una tarea a un grupo seguidores, cada uno propone una oferta para llevarla a cabo, y el líder elige llevar a cabo la tarea con el que proponga la mejor oferta. En nuestro caso esto es innecesario, ya que gracias a la utilización del mecanismo CBR, la asignación de las tareas se hace siempre de una forma óptima, teniendo en cuenta las experiencias pasadas. Se lleva a cabo un proceso de aprendizaje social por facilitación, en donde se actualiza la base de conocimiento de la organización. El modelo de coordinación obtiene la nueva información por medio de "observación" de lo que sucede a los agentes miembros de la organización. Así, la sociedad aprende según la manera en que el entorno interactúe sobre los agentes (Figura 7).

Comparando los primeros dos modelos, el sistema mostraba una mejora sustancial al aumentar el número de tareas, haciendo que el tiempo llevado a cabo para la planificación y replanificación fuese menor (Figura 26).

Por otro lado, el SMA presentado en este trabajo utiliza THOMAS en un entorno dinámico junto con las capacidades de auto-adaptación que se han añadido. THOMAS puede ser comparada con otras plataformas disponibles actualmente para la creación de organizaciones de agentes. Todas ellas, deberían tener en cuenta aspectos como la estructura organizacional (soporte de grupo, topología, herencia de roles, interacciones y reglas sociales) y la dinámica de la organización (entrada y salida de agentes externos, creación de roles, ciclo de vida de los agentes, control de comportamiento, adaptación y coordinación). La Tabla 5 muestra una comparación de algunas de las plataformas más populares mostrando el soporte u omisión de estos aspectos.

En la tabla se muestran las plataformas junto con características básicas en la gestión de los agentes que poseen o no. En la primera línea se indican las plataformas estudiadas y en la primera columna las características entre las que se incluyen: modelado (de agentes, organización y servicios); posibilidad de utilización del modelo BDI en los agentes; propiedades taxonómicas (creación de grupos, estructuras, roles e interacciones); propiedades normativas (creación de reglas y normas); dinámica de la organización (participación de agentes, divulgación de normas, control de comportamiento, participación de organizaciones y adaptación); propiedades de coordinación (planificación, planificación óptima, replanificación y reorganización); simulación de agentes de forma gráfica (2D y 3D). En la notación empleada en

la tabla, "\*" indica presencia de propiedad y "\*\*\*" presencia con algún módulo adicional que se incorpora a la arquitectura y que queda reflejado.

Tabla 5. Architectures comparison

Platforms\Features	JADE <sup>17</sup>	AMELI <sup>18</sup>	MADKIT <sup>19</sup>	JACK TEAMS <sup>20</sup>	S-MOISE <sup>+21</sup>	AEI	RETSINA	THOMAS
<b>Model</b>	*	**ISLANDER <sup>22</sup> (EI)		**JACOB <sup>23</sup>		**ISLANDER	*	**GORMAS <sup>24</sup>
<b>Agent</b>								
<b>Organization</b>	*	**ISLANDER(EI)		**JDK <sup>25</sup>		**ISLANDER	*	**GORMAS
<b>Services</b>	*	**ISLANDER(EI)	**26	**JDK		**ISLANDER	*	**GORMAS
<b>BDI Model</b>	*			*		**aBuilder	*27	*
<b>Group</b>			*	*	*	**AMELI	*	*
<b>Topology</b>		(EI) <sup>28</sup>	HOLONS	**29	*	**AMELI	*	*
<b>Roles</b>		*	*	*	*	**AMELI	*	*
<b>Interactions</b>	*	*	*	*	*	**AMELI	*	*
<b>Norms</b>		*	**JESS <sup>30</sup>	*	*	**AMELI	*	*
<b>Agent Joining</b>		*	*	*	*	**AMELI	*	*
<b>Role Enactment</b>		*	*	*	*	**AMELI	*	*
<b>Behavior control</b>	*	*	*	*	*	**AMELI	*	*
<b>Org. Joining</b>	*	*	*	*	*	**AMELI	*	*
<b>Adaptation</b>		*	*	*	*	**AMELI	*	*
<b>Coordination</b>				**JPL <sup>31</sup>		*	HTN <sup>32</sup>	*
<b>Optimms Planning</b>								*
<b>Replanning</b>							HTN	*
<b>Reorganization</b>					*			*
<b>Simulation 2D</b>	**		*	*		**Simdei <sup>33</sup>		**
<b>Simulation 3D</b>	**					**Simdei		**

\* Provides that feature

\*\* Provides this feature with some additional module

<sup>17</sup> JADE: <http://jade.tilab.com/><sup>18</sup> Electronic Institutions [Esteva:2003]<sup>19</sup> Madkit: [www.madkit.org](http://www.madkit.org)<sup>20</sup> JackTeams: [www.agentsoftware.com/sharedemos/ndocs/JACK\\_Teams\\_Manual.pdf](http://www.agentsoftware.com/sharedemos/ndocs/JACK_Teams_Manual.pdf)<sup>21</sup> [Hubner et al., 2006]<sup>22</sup> ISLANDER (Electronic Institutions Editor)<sup>23</sup> JACOB (Jack Object Modeller) <http://www.aosgrp.com/products/jack/registration/regform.html><sup>24</sup> GORMAS [Argente, 2008]<sup>25</sup> Jack Development Kit<sup>26</sup> Not specified architecture, it is possible to extend the BDI model.<sup>27</sup> RETSINA is an open MAS system that provides infrastructure for different types of deliberative, goal directed agents. In this sense, the architecture of RETSINA exhibits some of the ideas of BDI agents.<sup>28</sup> Topology of Electronic Institutions<sup>29</sup> Team Structure (Formation of the team)<sup>30</sup> Possibility of extension with Jess (a rule based engine)<sup>31</sup> JACK Plan Language (JPL) Jack provides an agent-specific plan language for writing JACK Plans<sup>32</sup> RETSINA Planner<sup>33</sup> SIMDEI allows to run discrete event simulations of AMELI.

THOMAS proporciona una ventaja clara respecto a sus coetáneas. Además, junto con el modelo social de planificación, proporciona los aspectos de reorganización y adaptación necesarios en entornos dinámicos como el propuesto.

Quizá la aproximación más similar al modelo propuesto se presente en las investigaciones llevadas a cabo sobre RETSINA (*Reusable Environment for Task-Structured Intelligent Network Agents*) [Giampapa y Sycara, 2002], donde se propone un modelo de coordinación social basado en planificación HTN (*Hierarchical Task network*) intentando revisar los planes llevados a cabo por la organización en tiempo de ejecución. Aún así, comparando las dos propuestas vemos cómo el modelo aquí presentado proporciona algunas ventajas que el primero no incluye. El modelo propuesto tiene la capacidad de reorganizar la sociedad en tiempo de ejecución, además de replanificar, también en tiempo de ejecución; pero además es capaz de obtener la ruta óptima para su organización, la ruta más fácilmente replanificable y aprender de las experiencias obtenidas a lo largo del ciclo de vida de la organización gracias al mecanismo de razonamiento basado en casos incluido.

Siguiendo este concepto, actualmente se están llevando a cabo investigaciones sobre la aplicación de técnicas de razonamiento y aprendizaje al modelado de organizaciones de agentes. En concreto en [Bou et al., 2008] [Bou et al., 2009] encontramos una propuesta en la que Instituciones Electrónicas Automáticas (AEI) se combinan con CBR para resolver situaciones que ya han sucedido en el pasado, pero en este caso, el aspecto de adaptación y reorganización dinámica no está contemplado.

Como conclusión a los resultados realizados, decir que se han conseguido los objetivos que se pretendían alcanzar: (i) Desarrollo de la sociedad de agentes; (ii) Simulación del comportamiento de la organización en un caso concreto de coordinación y adaptación de sus agentes; y (iii) validación del modelo de planificación propuesto mediante la simulación de la organización del caso de estudio.

Por último, el modelo presentado sobre la arquitectura THOMAS, al haber sido planteado con carácter general, no presenta restricciones especiales respecto a su dominio de aplicación; apostar por su adaptabilidad a problemas específicos no supone ninguna disminución en la eficiencia de las aplicaciones que se desarrollen.

---

## 7.2 CONCLUSIONES

---

Debido a los avances tecnológicos de los últimos años, la "sociedad" en la que participa un sistema multi-agente cada vez necesita más integrar requisitos como distribución, evolución, flexibilidad de entrada y salida de miembros,

gestión adecuada de la estructura organizativa, etc. Todos estos requisitos definen un conjunto de características que se pueden abordar mediante el paradigma de los sistemas abiertos y de las organizaciones virtuales. En esta tesis se ha presentado un nuevo modelo de coordinación para organizaciones de agentes abiertas. La principal novedad del modelo consiste en la capacidad de planificación dinámica y adaptativa para llevar a cabo una distribución óptima de las tareas de los agentes miembro de una organización.

Este capítulo describe cómo se han alcanzado los distintos objetivos definidos en esta investigación para validar y evaluar la hipótesis de partida: *“En el marco de este proyecto de investigación se probará que es posible modelar un sistema eficiente para la gestión de sistemas abiertos a partir de un modelo de organizaciones adaptativo que facilite una planificación distribuida y dinámica”*.

Consideramos que los requerimientos más importantes para que un sistema pueda considerarse **auto-adaptativo** [Di Marzo et al., 2004] y **abierto** [Dignum, et al., 2001] han sido cubiertos:

- ❖ Nos encontramos ante un sistema donde la reorganización ocurre como resultado de una planificación de un control central interno que utiliza mecanismos basados en interacciones directas y cálculos locales para reorganizar los agentes, y así obtener un estado global coherente del sistema.
- ❖ La organización de agentes incluye formalismos para la descripción, la construcción y el control de elementos organizativos y normativos en lugar de sólo los estados de los agentes [Artikis et al, 2001], [Zambonelli et al., 2001 ]. Además, la descripción de estos elementos son especificados de forma explícita desde una sociedad abierta; en la que no se depende de las intenciones, deseos y creencias de cada agente [Dellarocas, 2000], [Ossowski, 1998]. En este trabajo, debido a la utilización de THOMAS, se tiene el control sobre los miembros de la organización, coordinando las tareas y la toma de decisiones. Todos los miembros colaboran entre ellos para alcanzar un objetivo común, compartiendo su información, y la coordinación que surge mediante planes y decisiones.
- ❖ La metodología utilizada proporciona mecanismos para describir el entorno de la sociedad y las interacciones entre los agentes y la sociedad, y para formalizar el resultado esperado de funciones a fin de verificar la animación general de la sociedad. Proporciona directrices sobre la capacidad de comunicación y la capacidad para ajustarse al comportamiento que se espera de los agentes que participan en la sociedad. Se dispone de métodos [Argente, 2008] para poder comprobar si el diseño de una sociedad satisface sus necesidades y sus objetivos [Jonker et al., 2000].
- ❖ La elección del modelo de coordinación depende del ámbito de actuación de la propia organización, pero siempre debe intentarse

cubrir todas las posibilidades de actuación de los agentes. En este sentido, el modelo de coordinación para organizaciones propuesto en este estudio cubre todos los tipos de coordinación investigados (sección 3.2.1) ya que la organización determina globalmente las acciones de los agentes (coordinación global) pero dejando libertad a cada uno de ellos para que decida cómo resolver sus propios planes (coordinación individual). Además, la coordinación de la organización (a nivel *macro*) estará *orientada a problemas*, previniendo inconsistencias entre planes de cada agente y haciendo que cada uno se coordine a nivel de acciones cuando de manera global se decida el plan de la organización (a nivel *micro*).

La arquitectura THOMAS, utilizada junto al modelo para organizaciones propuesto, continúa en fase de desarrollo, sin embargo, los resultados preliminares demuestran que es aplicable al desarrollo de SMA abiertos basados en organizaciones con funcionalidades complejas, como es el caso del sistema *TouristOrg*. Específicamente, el modelo propuesto proporciona a la arquitectura los servicios necesarios para llevar a cabo una planificación de tareas de manera dinámica. Cabe destacar que, tanto el modelo como las tecnologías presentadas en esta investigación, son tan solo un ejemplo del potencial para desarrollar sistemas basados en el paradigma de las organizaciones virtuales.

---

### 7.3 CONTRIBUCIONES A LA INVESTIGACIÓN

---

El trabajo de investigación presentado en esta memoria aporta algunas nuevas contribuciones, principalmente en los ámbitos de la Teoría de Agentes y de las Organizaciones Virtuales. A continuación, se describen las principales contribuciones de esta investigación.

- ❖ Marco para el análisis y diseño de organizaciones de agentes. Desde el punto de vista de la ingeniería se ha realizado una revisión de las herramientas para el análisis y diseño de organizaciones, tratando de utilizar las ventajas de cada una de ellas. GORMAS es la metodología elegida en este trabajo. Es una guía metodológica para el diseño de SMA abiertos desde la perspectiva de las organizaciones humanas. Sus características permiten cubrir el análisis, diseño de la estructura organizativa y diseño de la dinámica de la organización de forma eficiente.
- ❖ Marco para el desarrollo de organizaciones de agentes. Se ha realizado una revisión del estado del arte de los sistemas multi-agente desde un punto de vista organizacional junto con algunas áreas relacionadas entre las que destaca la coordinación y planificación de agentes. Estas tecnologías proporcionan una infraestructura robusta que permite a

los agentes comunicarse y coordinarse de forma distribuida y obtener información del contexto de forma automática, requerimientos fundamentales en este tipo de desarrollos.

- ❖ Estudio de tecnologías complementarias, incluyendo agentes virtuales inteligentes, sistemas de razonamiento basados en casos y aplicación de estas tecnologías en los SMA. Este estudio es un compendio completo para que nuevos investigadores dispongan de literatura básica para iniciar futuros proyectos en áreas relacionadas. Por otra parte, algunas de estas tecnologías han sido aplicadas en un entorno real, descrito en detalle en el caso de estudio.
- ❖ Propuesta de un modelo integrador en el que una arquitectura de agentes puede ejecutar acciones basándose en un modelo de planificación social en una organización de agentes. El pilar del modelo es el razonamiento basado en casos, es decir, apoyándose en experiencias similares ocurridas en el pasado, en contextos similares, la organización actúa. El modelo ha sido utilizado para crear un entorno inteligente en un escenario real y se han evaluado positivamente sus capacidades para la planificación de una organización de agentes.
- ❖ Aplicación del modelo a un problema real. Se ha definido un caso de estudio para comprobar la validez del modelo propuesto, el cual consiste en el desarrollo de TouristOrg, una organización de agentes enfocada a mejorar los servicios de planificación un sistema de guiado turístico. De esta forma, se ha podido demostrar que la arquitectura junto con el modelo son una alternativa viable para desarrollar sistemas multiagente abiertos y adaptativos sobre entornos altamente dinámicos.
- ❖ Modelado del contexto. Una parte fundamental en todo desarrollo de organizaciones es el modelado del contexto. Se ha realizado un modelado del sistema TouristOrg, presentado en detalle en el caso de estudio, describiendo cada una de las fases. Este modelo servirá como base para el desarrollo de posteriores proyectos y líneas de investigación afines.
- ❖ Intercambio de conocimiento. Durante el proceso de desarrollo de este trabajo se ha realizado un gran esfuerzo por obtener una retroalimentación de distintos investigadores y grupos de investigación en áreas relacionadas. El objetivo es robustecer esta investigación mediante el intercambio mutuo de ideas y conocimiento. Se ha puesto especial interés en difundir nuestras experiencias y los avances de esta investigación, desde sus etapas iniciales hasta su forma final, a través de diversas publicaciones<sup>34</sup> y la asistencia a conferencias, congresos, *workshops*, etc.

---

<sup>34</sup> <http://bisite.usal.es> [Último acceso 9/2/2010]



La investigación presentada en esta memoria es relevante sobre todo desde el punto de vista científico, sentando las bases para el desarrollo de importantes proyectos; en especial, el proyecto THOMAS y el proyecto OVAMAH, ambos de ámbito nacional:

- ❖ Proyecto THOMAS<sup>35</sup> (“THOMAS: Métodos Técnicas y Herramientas para Sistemas Multiagente Abiertos”) TIN2006-14630-C03-03, otorgado por el Ministerio de Ciencia e Innovación
- ❖ Proyecto OVAMAH<sup>36</sup>: (Organizaciones Virtuales Adaptativas: Mecanismos, Arquitecturas y Herramientas) TIN 2009-13839-C03-03, otorgado por el Ministerio de Ciencia e Innovación. Proyectos de Investigación Fundamental No Orientada.

Estos proyectos permitirán continuar con el desarrollo de esta investigación y aplicarla en escenarios todavía más complejos.

---

## 7.4 TRABAJO FUTURO

---

La investigación presentada en este trabajo de tesis doctoral deja abiertas una gran cantidad de puertas para un trabajo futuro. Importantes entidades, y grupos de investigación han mostrado su interés en participar en el avance y crecimiento de dicha investigación aportando sus conocimientos. Entre ellos, el grupo GIA<sup>37</sup>- Grupo de Inteligencia Artificial (Universidad Rey Juan Carlos de Madrid), GTI-IA<sup>38</sup> - Grupo de Tecnología Informática - Inteligencia Artificial (Universidad Politécnica de Valencia) y el propio grupo BISITE - Grupo de Biomedicina, Sistemas Informáticos Inteligentes y Tecnología Educativa (Universidad de Salamanca), por lo que el futuro de esta investigación resulta prometedor a medio y largo plazo.

El proyecto THOMAS, referenciado en la sección anterior, tiene como objetivo proveer tecnología basada en agentes/sistemas multi-agente necesaria para el desarrollo de organizaciones virtuales en entornos abiertos. Cabe destacar la experiencia acumulada por el consorcio formado por los tres grupos de investigación participantes en el desarrollo de este proyecto y los excelentes resultados obtenidos, con más de 30 publicaciones con índice de impacto. Todo ello llevó a proponer un nuevo proyecto fundamentado principalmente en la necesidad de continuar avanzando en la línea propuesta. Este proyecto es OVAMAH, cuyo objetivo es el de avanzar y aportar soluciones principalmente en los aspectos relacionados con la reorganización de organizaciones dinámicas de agentes, proponiendo:

---

<sup>35</sup> <http://www.thomas-tin.org> [Último acceso 9/2/2010]

<sup>36</sup> <http://bisite.usal.es/webisite/?q=es/node/2> [Último acceso 9/2/2010]

<sup>37</sup> <http://www.ia.urjc.es/cms> [Último acceso 9/2/2010]

<sup>38</sup> <http://users.dsic.upv.es/grupos/ia/ia.html> [Último acceso 9/2/2010]

- ❖ Definición y construcción de una plataforma de soporte multi-dispositivo que permita la gestión robusta y optimizada de organizaciones virtuales que incluyan mecanismos de adaptación.
- ❖ Desarrollo de propuestas de modelado que permitan la definición y el diseño de técnicas y métodos para poder alcanzar acuerdos de tipo reorganizativo en este tipo de sistemas.
- ❖ Definición y desarrollo de las extensiones necesarias, tanto a nivel de metodología como de herramientas, para el modelado y desarrollo de organizaciones adaptativas.
- ❖ Desarrollo de técnicas de descripción semántica de organizaciones virtuales que proporcionen la información para ser usada a nivel micro y a nivel macro.
- ❖ Desarrollo de técnicas/métodos para el descubrimiento de organizaciones virtuales con determinadas características, así como situaciones no deseadas y su adaptación automática.
- ❖ Estudio de modelos deliberativos orientados a la reconfiguración de organizaciones virtuales en sistemas ubicuos
- ❖ Estudio y adaptación de los modelos obtenidos para su ejecución en dispositivos ligeros, así como compatibilidad con tecnologías de conectividad inalámbrica más adecuadas para la implantación de este tipo de sistemas
- ❖ Estudio de técnicas de simulación social existentes y desarrollo de modelos de acuerdo a la arquitectura propuesta para la reorganización en organizaciones virtuales.
- ❖ Implementación de componentes software que instrumenten las propuestas anteriores.
- ❖ Desarrollo de herramientas “case” que permitan modelar organizaciones virtuales, así como los diferentes mecanismos deliberativos de reconfiguración propuestos
- ❖ Desarrollo de una herramienta de simulación, basada en tecnología Web3D, para la aplicación y evaluación de la tecnología desarrollada.
- ❖ Aplicación y evaluación de la tecnología desarrollada a través de demostradores software para casos de estudio reales.

La hipótesis de partida de este proyecto consiste en dotar de capacidades de autonomía a una organización virtual que permitan una respuesta dinámica frente a posibles situaciones cambiantes por medio de la adaptación y/o evolución de la propia organización, de tal manera que ésta sea capaz de detectar situaciones de interés, como por ejemplo fallos de funcionamiento, y pueda manejarlos maximizando la flexibilidad y la capacidad de adaptación.

Por lo tanto, el trabajo presentado en esta tesis es una contribución importante en los objetivos a corto plazo para este proyecto, ya que *se ha modelado un sistema de planificación capaz de adaptarse a los objetivos comunes de los agentes deliberativos que participen en una organización de agentes*. Las

posibles líneas de trabajo a seguir directamente relacionadas con este trabajo podrían ser:

- ❖ Mejora y desarrollo de las extensiones necesarias en la arquitectura THOMAS, tanto a nivel de metodología como de herramientas, para el modelado y desarrollo de organizaciones adaptativas.  
THOMAS, aún bajo desarrollo, permite la extensión y mejora de sus módulos. La arquitectura ha sido utilizada para la creación de una organización de agentes y se han evaluado positivamente sus capacidades para modelar casos de estudio reales como el descrito en el capítulo 6; pero dado que existen capacidades reorganizativas en este tipo de sistemas que aún pueden ser estudiadas y desarrolladas, parece conveniente hacerlo directamente como extensiones generales de THOMAS que sirvan para cualquier tipo de organización.
- ❖ Mejora del modelo propuesto.  
Tal y como se ha indicado en este trabajo, existen muchos enfoques para llevar a cabo tanto la planificación de tareas por parte de una organización como su adaptación. Parece interesante comparar el modelo propuesto frente a otros tipos de planificación, como las redes de contratos descritas en el apartado anterior, abriendo así un camino de investigación futuro para la mejora del mismo.
- ❖ Estudio y adaptación del modelo de adaptación obtenido en este trabajo para su ejecución en dispositivos ligeros, así como compatibilidad con tecnologías de conectividad inalámbrica más adecuadas para la implantación de sistemas ubicuos.  
Aunque dado la tecnología y los conceptos base empleados en este estudio, no tendría que haber ningún inconveniente, se considera adecuado abrir una línea de investigación en la que los modelos descritos puedan ser probados en dispositivos ligeros.
- ❖ Estudio de técnicas de simulación social existentes y desarrollo de modelos de acuerdo a la arquitectura propuesta para la reorganización en organizaciones virtuales.  
Tanto el modelado como la simulación de organizaciones abren dos frentes de investigación muy amplios. Por un lado, las tendencias actuales tienden a llevar a cabo la representación de agentes de manera visual con técnicas novedosas, como por ejemplo la tecnología Web3D, MMOG (Massively multiplayer online game), o los mundos virtuales, para la evaluación de los sistemas desarrollados. Por otro lado, la simulación del comportamiento de los agentes dentro de una organización, y su forma de adaptación frente a posibles cambios, abre también otros caminos de investigación en los que trabajar; por ejemplo, predicciones o análisis de patrones de comportamientos. Hemos visto como tecnologías como REPAST, empleada en este trabajo ofrece la posibilidad de utilización de modelos tridimensionales, estudios estadísticos en tiempo real, simulaciones dirigidas y utilización de herramientas como Matlab, Weka o VisAd, lo que

proporcionaría nuevas funcionalidades de desarrollo de algoritmos o estudios de predicción y minería entre otros. Este trabajo hace una pequeña inmersión en este mundo, dejando abiertas muchas puertas en estos dos sentidos.

❖ Pruebas y validación.

Es necesario realizar pruebas mucho mas exhaustivas con el objetivo de evaluar en detalle el modelo propuesto en términos de tiempo, aplicación de análisis y diseño, tiempo de cálculo de respuestas, calidad de las respuestas, etc. Los resultados obtenidos permitirán desarrollar modelos y sistemas más depurados y robustos.

❖ Resolución de nuevos problemas prácticos.

Para realizar una comprobación más exhaustiva de la validez del modelo propuesto, es necesario aplicarlo a nuevos problemas prácticos. De esta forma, sería posible comprobar si se adapta de forma adecuada a la resolución de problemas de características distintas, o bien si se ha construido un modelo muy restringido a los problemas concretos que se han estudiado durante la realización de este trabajo.

## PARTE V. MODEL OVERVIEW



## 8 MODEL OVERVIEW

---

This chapter is a summary of the research presented in this thesis work. The chapter is structured as follow. Section 8.1 is a brief introduction of the study. Section 8.2 describes the primary characteristics of the agent organizations and the state of the art for current studies in this field. Section 8.3 presents the structure of the proposed planning model, and ties it to the concepts of the CBP-BDI agent. Section 8.4 demonstrates how the model can be used in a specific case study; to this end, the context, the concepts and the model of the system will be explained. Section 8.5 will provide a clear example of planning within the developed system. And finally, Section 8.6 will emphasize the results and conclusions obtained.

---

### 8.1 INTRODUCTION

---

Services based on new technologies are generating new lines of investigation with respect to improving the interactive experience; eliminating any irrelevant services, or those that hold little or no interest, and adapting to the true needs of each user. One of the primary tendencies in a tourist setting is to implement systems to help tourists plan their routes. Recommendation and guidance systems are an attempt to mathematically model and technically emulate the real world recommendation process. These are recognized Artificial Intelligence techniques based on software components that are typical in e-commerce and tourist systems. CBCF (Content-Boosted Collaborative Filtering) [Melville et al., 2002], LIBRA (Learning Intelligent Book Recommending Agent) [Mooney and Roy, 2000], MRS (Music Recommendation System) [Hung-Chen and Arbee, 2005], or TIP (Tourist Information Provider) [Hinze and Voisard, 2003] [Hinze et al., 2005] are some examples.

MAS (Multi-Agent Systems) stand out among the different AI techniques currently employed, having been used successfully in these types of applications [Fonseca et al., 2001][Kowalczyk et al., 2002] [Bajo et al., 2009]. Because of their particular characteristics, including autonomy, flexibility and the ability to cooperate, they are perfectly suited for dynamic environments such as these. The current trend with MAS systems is to employ organizational concepts in their design, which helps to focus their work in the field of tourism, since they establish a connection between a tourist organization and an organization at the software agent level with regards to structure, functionality, etc.

Virtual organizations [Ferber et al., 2004] are a means of understanding system models from a sociological perspective, which is closely related to the area in question. From a business perspective, a virtual organization model is based on

the principles of cooperation among businesses within a shared network, and exploits the distinguishing elements that provide the flexibility and quick response capability that form the strategy aimed at customer satisfaction [Schertler, 1998]. It is therefore in the best interest of tourist agencies to begin taking into consideration this type of organization, which takes advantage of the efficiency of new information and communication technologies, to improve their competitiveness among other agencies.

Even so, within the development of organizations, both at the business and agent level, we find a set of requirements [Rodríguez et al., 2009] that call for the use of new social models in which the use of open [Zambonelli et al., 2003] and adaptive [Di Marzo et al., 2004] systems is possible. Open systems are characterized by the heterogeneous quality of their participants, limited trust factor, opposing individual objectives, and a high probability of discontent with the specifications [Giret et al., 2005]. Adapting to the environment is a goal that has been recognized in the most recent research [Zambonelli et al., 2004][Reitbauer et al., 2004][Weyns et al., 2004] [Capera et al., 2003][Razavi et al., 2005]. It involves software systems capable of responding and acting on their own accord when dealing with changes in their environment. This requires the availability of theories, models, mechanisms, methods and tools that can develop systems with the ability to reorganize and adapt to future changes in their environment. Even when focusing on the field of tourism, it is possible to see that adaptation is considered a common denominator within the research in this area: adaptation of (i) the offer (a specific tourist context, and several basic service requirements needed to perform this activity); (ii) the providers (the tourist information and the data base for available routs); (iii) the consumers/users (the user models). Each of these factors is critical when personalizing a tourist route adapted to the service that the tourist agency wishes to offer.

The tourists themselves are a fundamental part of the system. Those who are motivated to engage in a tourist route must choose where to go, how to arrive at their destination, and what to do once they arrive. The planning process for the route is complicated because of the numerous factors involved in this process, such as a preference for certain attractions, means of transport, services, as well as the time and money restrictions of each tourist. The process of creating personalized routes or itineraries based on the user's profile and certain restrictions, can be solved by applying techniques that are well known in Operational Research (OR). OR is an interdisciplinary science that applies scientific methods to solve daily problems within an organization, and seeks to improve efficiency. Some of the algorithms associated with the planning process are based on the Travelling Salesman Problem (TSP) (Travelling Salesman Problem [Held and Karp, 2005], Traveling Salesman Problem with Time Windows [Dumas et al., 1995], Generalized Travelling Salesman Problem (GTSP) [Labordere, 1969] Orienteering Problem (OP) [Golden et al., 1987], Chinese Postman Problem (CPP) [Thimbleby, 2003]) but these algorithms do



not incorporate any critical restrictions for the functional tourist system. Each of these solutions focuses on aspects related to mobility. Their objective is to find a route between a start and end point that maximizes a specific benefit and minimizes costs within a specific period of time for a single user. Although these solutions have been studied for a considerable amount of time, there is still no optimal solution for the problem of planning scalable routes to non-deterministic environments areas or those shared by several members of an organization. This problem poses a considerable calculation, and there does not yet exist a solution to finding a self-adaptive planning for the variation of points along the route.

Some of the algorithms that currently exist can be used to calculate routes within the tourism sector. The routes can be personalized based on the tourist's profile, the attractions and the information about the trip. Each route typically begins by planning the day very generally. This phase is important, although it would be much more useful to create a dynamic itinerary in which the system likewise offers a list of recommended attractions based on the interests of the users, the other users, and the possible replanning of the plans. The problem with self-adaptive planning within an organization can be extrapolated to other situations and daily life case studies (business, educational, entertainment, etc. environments).

*Given the advantages provided by the unique characteristics found in the development of MAS from an organizational perspective, and the absence of an adaptive planning process for any social model, this study proposes a model that can coordinate a dynamic and adaptive planning system in an agent organization. Using a combination of multi-agent technology and OR-based planning techniques, it is our goal to develop an optimal distribution of tasks for the member agents of an organization.*

The model is unique in its conception and can provide an organization with self-adaptive capabilities in execution time for highly dynamic environments. This allows the behavior of the agent to be determined by the goals it wishes to achieve, while taking into account the goals of other member agents as well as changes in the environment. In order to evaluate the model, a case study was carried out within the tourist sector, which has always strived to optimize the process of generating plans and distributing resources. The case study designed for this research project has significant bearing. However, the objective of the research project itself is not to provide a dissertation on behavior within a tourist environment, but to validate the proposed planning model through the simulation of an organization within this sector.

## 8.2 STATE OF THE ART

---

The usefulness of any technology, including multi-agent systems (MAS) can be judged by two criteria: (i) its ability to solve new types of problems and (ii) its ability to improve the efficiency of existing solutions [Jennings, Wooldridge, 1998]. With this in mind, agents and multi-agent systems provide a natural method of characterizing intelligent systems. Intelligence and interaction are two concepts that are inextricably joined, a fact that is well established in agent technology.

When discussing MAS, the idea of a single agent is expanded to include an infrastructure for interaction and communication. Ideally, MAS include the following characteristics [Huhns, Stephens, 1999]:

- ❖ They are typically open with a non-centralized design.
- ❖ They contain agents that are autonomous, heterogeneous and distributed, each with its own “personality” (cooperative, selfish, honest, etc.).
- ❖ They provide an infrastructure specifically for communication and interaction protocols. Open MAS should allow the participation of heterogeneous agents with different architectures and even different languages [Zambonelli et al., 2003]. However, this makes it impossible to trust agent behavior unless certain controls based on norms or social rules are imposed. To this end, developers have focused on the organizational aspects of agent societies, using the concepts of organization, norms, roles, etc. to guide the development process of the system.

MAS agents based on organizational concepts coordinate and exchange services and information; they are capable of negotiating and coming to an agreement; and they can carry out other more complex social actions. At present, research focusing on the design of MAS from an organizational perspective seems to be gaining most ground. The emergent thought is that modeling the interactions in a MAS cannot be related exclusively to the actual agent and its communication capabilities; instead, organizational engineering is necessary as well. The concepts of rules [Zambonelli, 2002], norms and institutions [Esteva et al., 2001] and social structures [Parunak, Odell, 2002] are rooted in the idea of needing a higher level of abstraction, independent from the agent, that explicitly defines the organizations in which the agents reside.

Coordination and communication are very important themes within MAS. Over the years, many different types of architectures and models to manage the organization and coordination in a MAS have been proposed [Wooldridge y Jennings 1994][Wiederhold, 1992]. All of the architectures have one point in common: the concern for coordinating agent behavior. Agents can jointly

coordinate their knowledge, goals, abilities, tasks and plans to resolve a global objective. To perform tasks, an agent may need system resources, in which case it would have to coordinate with the other system agents that need the same resource. The coordination allows the agents to consider all of the tasks and regulate them so as to avoid executing undesirable actions such as:

- ❖ The agents not generating or communicating sub-solutions that would lead to the solution of a problem.
- ❖ The agents generating and communicating redundant solutions.
- ❖ An inappropriate distribution of the workload among agents.

This coordination is related to the action planning required for solving tasks:

- ❖ High level understanding and ability to predict the behavior of the other system agents.
- ❖ Exchange intermediate results that progress towards the solution of a global task.
- ❖ Avoid redundant actions, if they are undesirable.

Having the agents carry out tasks can be organized in various ways, such as: each agent executing one of the tasks; or dividing the tasks into sub-tasks, which can be done by a breakdown mechanism for problems, that are performed by various agents. The tasks that an agent should perform depend, among other things, on the role that the agent assumes in the system. For example, in an “office” system, an agent “person” assumes the role of “secretary” and performs all duties related to that role. That same agent could assume the role of “boss” and perform very different tasks related to this new role. To carry out these tasks, an agent may need system resources, in which case it must coordinate with the other system agents that need to use the same resource.

Agent organizations depend on the type of coordination and communication among agents, as well as the type of agents that comprise the group. There are several different organizational approaches: MOISE [Boissier 2007], OperA [Dignum 2004], INGENIAS [Pavon 2003], the Gaia Methodology [Zambonelli et al., 2003], [Omicini et al., 2004]. However, while these studies provide mechanisms for creating coordination among participants, there is much less work focused on adapting organizational structures in execution time, or on norms defined in design time. For example, [Hubner 2004] proposes a model for controlling adaption by creating new norms. [Gasser & Ishida 1991] propose a distributed model for reorganizing their architecture. [Artikis et al. 2009] requires agents to follow a protocol to adapt the norms. Each of these studies focuses on the structure and/or norms based on adapting the coordination among participants.

There are other approaches, such as those based on social norms (on the role of a social group and the effect on the appearance and support of social norms) [Hales 2002][Lakkaraju y Gasser, 2008].

Another possibility is the development of a MAS that focuses on the concept of organization/institution. Human interactions are regulated by institutions [North 1990] that represent the rules of the game in a society, and define what the individuals can and cannot do and under what conditions. This perspective defends the adoption of a mimetic strategy whose objective is to tackle the complexity of the development of open MAS from an organizational concept. One electronic institution [Noriega 1999][ Rodríguez-Aguilar, 2003][Esteva 2003][Garcia 2009] should be considered a social middleware between the external participating agents and the selected communication layer responsible for accepting or rejecting the agent actions. There are also different studies that focus this theme on the definition of “Automatic Electronic Institutions”, an extension of electronic institutions with self-adaptive capabilities [Bou et al.,2007a][Bou et al.,2007b][Bou et al., 2009]. The primary difference with the other proposals is that the adaption is carried out by the institution instead of by the agents.

Other points of view [Shoham, 1992][Shoham, 1997][Kittock, 1993][Walker Wooldridge, 1995] focus on the mechanisms for regulating behavior. Here the key factor is “imitation”, which refers to the social conventions that occur through propagation and contagion [Conte & Paolucci, 2001][Burt 1987].

Lastly, there are approaches in which the agents use the information obtained during their interactions [Axelrod, 1986][Castelfranchi, 1995][Sen, 2007]. In these studies, social norms (as special types of social conventions) are used by the agents to self-regulate their behavior according to social information: prior history, reputation, etc. The research focuses on social group mechanisms based on the social information gathered during the interactions [Villatoro y Sabater-Mir, 2008a][Villatoro y Sabater-Mir, 2008b][Villatoro y Sabater-Mir, 2008c]. They study the effects of the information transmission algorithms in recognizing and forming social groups.

None of these approaches is capable of coordinating tasks for the member agents of the organization to solve a common problem, nor do they consider that task planning should adapt to changes in the environment. The social model used in the architecture selected for this study is THOMAS (MeTHods, Techniques and Tools for Open Multi- Agent Systems) [Carrascosa et al., 2009] [Giret et al., 2009], which focuses on defining the structure and norms. The adaptation feature used in these models is based on the coordination among organization participants. The following section will present the proposed coordination model whose goal is to carry out an adaptive planning process within an agent organization.

---

### 8.3 PROPOSED MODEL

---

This thesis work proposes a coordination model that facilitates a self-adaptation feature within an agent society. We will use a cooperative MAS in which each agent is capable of establishing plans dynamically in order to reach its objectives. Additionally, there should be a global mechanism that can optimally assign activities to the agents so that they can work in a coordinated effort. The global mechanism considers the global objective of the society, as well as its norms and roles.

The proposed model clearly demonstrates that the suppositions set forth in section 8.2 do not occur. The agents have limited capabilities, which makes it difficult for them to have complete and accurate knowledge about the environment. Furthermore, the effects of the agents' actions can vary for different reasons (changes in the internal structure of the agents, changes in the norms, etc.). The behavior of the agents will be determined by the individual goals that each agent wishes to obtain, but while also bearing in mind the goals of other agents and any changes in the environment. Section 8.3.1 presents the deliberative CBP-BDI agent architecture proposed to construct the model, as well as the planning model used by the CBP-BDI agent to reach its individual objectives. Section 8.3.2 details the global coordination mechanism used in the study.

---

#### 8.3.1 CBP- BDI Agents

---

Among the different agent architectures that attempt to provide a satisfactory solution to these requirements, there is one model of particular interest that is based on the notion of deliberative reasoning [Bratman et al., 1988] [Corchado 2004] and has given rise to the BDI (Belief-Desire-Intentions) architecture. A BDI agent emphasizes the reasoning process directed at an action, i.e., determining what the agent should do [Wooldridge 2002]. This process involves two fundamental activities: a) determining which goals should be achieved (deliberation) and b) deciding how to reach these goals (planning). Both processes should be carried out by taking into account the limited resources of each agent. In other words, the reasoning processes should consider the restrictions imposed within the system for selecting the actions in an effective manner.

BDI architectures exhibit interesting properties that can deal with situations involving high level coordination among the agents available at a given time, and suggest the necessary extensions for this model, thus allowing the agents to engage in the correct social behavior.

The model proposed in this thesis includes a type of agent capable of planning from the perspective of an agent organization. This particular agent is a deliberative CBR-BDI agent [Corchado y Laza, 2003]. The primary characteristic of this agent is the ability to integrate a CBR reasoning mechanism [Admond y Plaza, 1994]. A CBR system initiates a reasoning cycle comprised of four sequential phases: retrieve, reuse, revise and retain.

This cycle is integrated within the activities of the BDI agent [Corchado y Laza, 2003] and identifies the phases as tasks or roles that the agent should be able to perform. This makes up for one of the primary deficiencies in the BDI model, which involves the manner in which memory and past experiences are handled. [Bajo et al., 2008], [Carrascosa et al., 2008], [Corchado et al., 2008], [Glez-Bedia, 2004] present a method for incorporating a CBR engine to the BDI model. The main idea in these studies is the use of the mechanisms that the deliberative BDI model provides, specifically Beliefs, Desires and Intentions, to be able to obtain a representation of the case and initiate the CBR reasoning cycle. To integrate the CBR reasoning system within the structure of a deliberative BDI agent, [Corchado et al., 2008], and [Glez-Bedia, 2004] propose a formula that can relate the concept of case to the fundamental concepts of BDI.

One specialization of the CBR-BDI agent is the CBP-BDI agent [Corchado et al., 2008]. Its internal structure is similar, but it differs from the CBR-BDI agent in that the solution that the agent carries out is not a model, but an actual plan, i.e., a sequence of actions. A CBP-BDI agent should incorporate a planning mechanism that generates plans as solutions to the given problem. A CBP-BDI agent can incorporate different planning mechanisms [Bajo et al., 2007], [Bajo et al., 2008], [Corchado et al., 2007], [Corchado et al., 2008]. [Corchado et al., 2008] presents a replanning mechanism for CBP-BDI agents in which the main characteristic is the ability to replan in execution time. A CBP-BDI agent calculates the plan or intention that is most easy to replan: Most RePlannable Intention (MRPI). This is the plan that can most easily be replaced by another plan in case it is interrupted (for example, if a user changes preferences while the plan is being executed).

In a dynamic environment, such as an agent society, the ability to have alternate plans is very important for maintaining efficiency within the system. As previously mentioned, it is increasingly common to model a MAS not only from the perspective of the agent and its communication capabilities, but by including organizational engineering as well. The primary function of a society is to allow its members to coexist in a shared environment and to carry out their respective goals while cooperating, or not, with the other members. The coordination of tasks that are combined to solve a common problem requires either a centralized or distributed type of planning that can be carried out by the system agents themselves. This constitutes an open problem for which traditional planning systems are inadequate. It is necessary to have a planning feature that can adapt to these circumstances.

---

### 8.3.2 Planning model for a CBP-BDI agent

---

The main objective of this study is to obtain a planning model that can, within an architecture geared towards the development of agent organizations, (THOMAS [Giret et al. 2009][Carrascosa et al. 2009]), take into account the changes that are produced within an environment during the execution of a plan. The particular feature that must be considered in this case is that the plans are executed by the member agents of an organization. The plans should take into account not only the restrictions imposed by an individual agent, but also the commitments, restrictions and norms within the organization. The planning process should define the actions that the society of agents will have to execute and should therefore also take into account the particular circumstances of each of its members. Essentially, this is an adaptive planning method within an agent organization. To achieve this, a CBP-BDI agent, as previously described, must be used, applying the following planning model that is particularly suited for organizations.

A plan  $p$  within an organization is defined as  $p = \langle E, O, O', R, R' \rangle$ , where:  $E$  is the environment that represents the type of problem that the organization solves, and is characterized by a set of states  $E = \{e_0, e^*\}$  for each agent, where  $e_0$  represents the initial state of the agent when the plan begins, and  $e^*$  is the state or set of states that the agents tries to achieve.  $O$  represents the set of objectives for the individual agent and  $O'$  is the set of objectives reached once the plan has been executed.  $R$  is the set of available resources for the given agent and  $R'$  is the set of resources that the agent has used during the execution of the plan.

Given the initial state of the organization, the term *global planning* is used to describe the search for a solution that can reach the final state, all the while complying with a series of requirements for the organization. The problem can be represented in a planning space that is delimited by the restrictions imposed by the requirements.

Given a common objective, specified resources available and tasks to perform, the aim is to find a global plan that allows the organization to find the optimal solution. To this end, the planning agent should bear in mind the optimal plans  $p^*(t)$  obtained for each individual agent. It is not necessary for all of the agents within the organization to know how to meet the objectives, but they should know how to perform some of the tasks that contribute towards reaching those objectives for the organization. Various generic plans are generated to solve a problem. They will be subsequently broken down into different tasks that will be distributed among the agents in the group (Figura 10).

Upon initiating the process, certain agents will be retrieved from the memory of cases to perform at least one of the problem tasks. For each task that is not completed by any of the retrieved agents, at least one new agent will be incorporated. This agent will have the greatest probability of successfully

completing the given task. The idea is to count on the necessary agents so that no task is left unassigned; that is, that grouping the tasks that the agents can perform should at the very least involve the set of tasks that must be performed in order to carry out the plan.

Let us assume that the common objective for agents “m” has “n” states or tasks with  $m, n \in \mathbb{N}$ .

Each agent has its own characteristics with regards to which tasks it can perform, which resources to use, and the amount of time available to perform the tasks. In other words, each agent has its own profile.

Given a state “j” for each agent “i” where  $i \in \{1, \dots, m\}_{m \in \mathbb{N}}$ , it can be defined with a tuple  $z_{i,j}$  - where each coordinate in the tuple refers to the characteristic that defines it.

The following binary variables are defined as:

$$a_{ij} = \begin{cases} 1 & \text{if agent "i" is assigned to task "j"} \\ 0 & \text{otherwise} \end{cases} \quad (8.1)$$

For each problem related to assigning tasks, an objective function is defined whose goal is to minimize and maximize the cost used by agents “m” to perform the common objective. For example, minimize or maximize the cost of using one of the agents to reach an objective, or maximize an efficiency function as need for each case.

A new efficiency function is introduced in order to assign tasks to the agents. Its aim is to visit the greatest number of points with the lowest possible cost. The aim of the previous efficiency function [Corchado et al, 2008], was to find, after having first assigned the tasks to the agents, the best possible route for completing the plan. The new efficiency function depends on the number of points visited and the total cost that this implies.

Cost is another function that depends on the time that agent “i” has spent working on task “j”, on the resources used, and on the type of agent assigned to each task. This is represented as:  $C_{i,j}^i$ .

The efficiency function is defined as:



$$Efficiency = \frac{N^{\circ} \text{ points visited}}{\sum_{i=1}^m \sum_{j=1}^n c_{t_{ij}r_{ij}}^i \cdot a_{ij}} \quad (8.2)$$

Let us assume we want to maximize the efficiency function:

$$Max \frac{N^{\circ} \text{ points visited}}{\sum_{i=1}^m \sum_{j=1}^n c_{t_{ij}r_{ij}}^i \cdot a_{ij}} \quad (8.3)$$

where  $t_{ij}$  is the time it takes agent "i" to perform the task, and  $t_{ij} = \text{Max}_k \{ t_{ijk} \}$  where  $t_{ijk}$  indicates the time it takes agent "i" to perform task "j" for tourist "k". Taking the maximum value of "k" (type of tourist), we can ensure that the guide has time to perform the necessary task regardless of the type of tourist. These times are initially estimated.

Let us now define the restrictions of the problem.

1.- We want each state to be completed by an agent, which in mathematical terms can be stated, for each state "k" as:

$$\sum_{i=1}^m a_{ik} = 1 \quad \forall k \in \{1, \dots, n\} \quad (8.4)$$

2.- We want each state to be completed within a specified period of time. Let us assume that state "k" should be completed within time  $t_k$ . The restriction would be:

$$\sum_{i=1}^m t_{ik} a_{ik} \leq t_k \quad \forall k \in \{1, \dots, n\} \quad (8.5)$$

3.- Each state "k" needs a set of resources to be executed. There is no reason for all of the agents to have these resources.

Given state "k", we need  $r_h^k$  resources with  $h \in \mathbb{N}$ , where  $r_w = \text{max}_{h \in \mathbb{N}} \{ r_h^k \}_{k=1, \dots, n}$ .

The variables  $\{ r_x^k \}_{x \in \{1, \dots, w\}} \quad \forall k \in \{1, \dots, n\}$  are defined in binary form:

$$r_x^k = \begin{cases} 1 & \text{if the agent "k" needs the resource "x"} \\ 0 & \text{otherwise} \end{cases} \quad (8.6)$$

The agent that performs state “k” must at the very least have at its disposal the resources that are needed to perform state “k”, for which, given state “k”, for each resource from the set  $\{r_x^k\}_{x \in \{1, \dots, w\}} \quad \forall k \in \{1, \dots, n\}$  we can define the following restriction:

$$\sum_{i=1}^m r_{ix} a_{ik} \geq r_x^k \quad \forall k \in \{1, \dots, n\}, \forall x \in \{1, \dots, w\} \quad (8.7)$$

The variables  $\{r_{ix}\}_{x \in \{1, \dots, w\}} \quad \forall i \in \{1, \dots, m\}$  are binary variables:

$$r_{ix} = \begin{cases} 1 & \text{if the agent "i" has the resource "x"} \\ 0 & \text{otherwise} \end{cases} \quad (8.8)$$

4.- Each agent “i” has a minimum and maximum time for work, depending on the type of agent. These times are represented as

$$t_i^{Turn\ on} \text{ and } t_i^{Turn\ off} \text{ respectively.}$$

$$t_i^{Turn\ on} \leq \sum_{j=1}^n t_{ij} \leq t_i^{Turn\ off} \quad \forall i \in \{1, \dots, m\} \quad (8.9)$$

For the majority of agents, as we will see in the case study, the maximum number of working hours is equal to a regular 8 hour work day.

5.- Every time we assign tasks to an agent, we want it to perform the minimum number of tasks, which varies according to the type of agent.

$$\sum_{j=1}^n a_{ij} \geq NumberTask_i \quad \forall i \in \{1, \dots, m\} \quad (8.10)$$

If the suggested problem of non-linear programming were incompatible, we would add agents to make it compatible. The agent added would be the one with the highest probability a priori of performing the necessary tasks.

If a norm (restriction) changes, it would be necessary to assign tasks once again.

This allows us to obtain a plan for the tasks that need to be performed by the agent organization. In other words, we can obtain a global plan composed of all the tasks and agents in the organization that will carry them out. Every agent in the organization recognizes the tasks that it needs to perform. These agents, which are CBP-BDI agents, integrate the 4 phases of a CBR system (retrieval, reuse, revise and retain). In each of these phases, the case-based planning performs the following actions [Corchado et al., 2008]:

- ❖ Retrieval phase: Select the plans with the information most similar to the problem case.
- ❖ Reuse phase: During this phase the agent creates a hyper-dimensional vector space that includes all of the restrictions for the problem, using the *B-spline* technique. At this point, all of the plans meeting the user's requirements are identified. This process involves geodesic curves (in the hyper-dimensional vector space), which can be calculated using the Euler theorem. Geodesic plans ensure a minimum risk and constant efficiency, and they satisfy the objectives of the agent. The minimum Jacobi field identifies the plan with the greatest density of plans surrounding it. This implies that if the plan is interrupted, it is possible to find another plan close by that should be able to reach the initial objectives. The CBP-BDI agent recommends the most easily replannable plans to the user.
- ❖ Revise phase: During this phase the user confirms the plan.
- ❖ Retain phase: During this phase the plans are stored according to the efficiency obtained.

This way it is possible to obtain a planner that focuses on providing the most optimal and efficient plan with minimal risk, in case of interruption, for a CBP-BDI agent. The planning problem considered is confined by certain restrictions.

To solve a constraint satisfaction planning problem, if we start with a set of states or tasks  $E = \{e_0, \dots, e_n\}$  that must be completed, it is possible to define an individual plan, such as a sequence of actions that start from state  $e_0$  and define the sequence of states through which an agent needs to pass in order to provide the most optimal completion of its tasks:

$$a_j : E \rightarrow E \quad (8.11)$$

$$e_j \rightarrow a_j(e_j) = e_{x_j}$$

The dynamic relationship that is established between an agent's behavior and the changes that are produced in its surrounding environment can be modeled through the following equation, which indicates that the behavior of an agent can be represented by its function of action  $a_A(t) \forall t$ , which is defined as a connection between the instance of time  $t$  and the action selected by the agent.

$$\text{Agent } A = \{a_A(t)\}_{t \in T \subseteq N} \quad (8.12)$$

Based on the definition of the function of action, it is possible to define a new relationship that represents the idea of a plan of action for an agent.

$$p_A : \begin{matrix} TXA & \rightarrow & A \\ (t, a_A(t)) & \rightarrow & p_A(t) \end{matrix} \quad (8.13)$$

in the following way:

$$p_A(t_n) = \sum_{i=1}^n a_{iA}(t_i - t_{i-1}) \quad (8.14)$$

This makes it possible to obtain a discrete time expression for a plan of action for an agent. However, the objective of the CBP-BDI agent is to obtain a dynamic planning system. Therefore, to obtain the definition of a plan for a CBP-BDI agent, we must present a continuous time extension for the expression proposed in the equation. In other words:

$$p_A(t_n) = \int_{t_0}^{t_n} a_A(t) dt \quad (8.15)$$

The environment in which a plan is executed is dynamic, and the plan for an agent  $p_A(t)$  can vary. Additionally, according to the CBP-BDI model, a plan can also be modified according to the known success or failure of plans that were previously executed. All of the plans have an associated efficiency function defined by a measure that can value the soundness of a plan according to its initial objectives, the resources used to obtain the objectives, and the results obtained after executing the plan. The efficiency expression for a plan is defined by the relationship between the objectives reached and the resources consumed.  $O$  indicates the initial objectives for the plan and  $O'$  represents the objectives reached after executing the plan.  $R$  is the total amount of resources available for executing the plan and  $R'$  represents the resources consumed by the agent during the plan's execution. The operator  $\#$  represents the cardinal point of a set.

$$E_{ff} = \frac{\#(O' \cap O)}{\#R'} \quad (8.16)$$

The planner should provide plans of optimal efficiency and minimal risk; it should search among the optimal efficiency plans for those with the best possibility of replanning in case of an interruption [Corchado et al., 2008].

Given an initial state  $e_0$  for an agent, the term individual planning is used to describe the search for a solution that enables reaching the final state  $e_i \equiv e^* \in E$  that satisfies a set of requirements. At this point we have a plan  $p^*(t)$ , which enables an agent to reach its objectives as optimally as possible, and subsist in a dynamic environment.  $p^*(t)$  provides all of the states through which an agent must pass in order to carry out a solution considered to be the most optimal for completing the defined tasks, while bearing in mind the resources available [Corchado et al., 2008]. We can obtain a structure for a CBP-BDI planning agent with dynamic attributes, i.e., the agent considers the possibility that there are changes that take place during the execution of a plan.

When introducing a problem to be solved in the organization, it will be necessary to take all restrictions into account. For example, only certain agents can carry out plans that arrive at the optimal solution. Once those agents have been identified, it will be necessary to assign the actions that each agent will carry out, and to generate a *global plan* that will indicate the states through which it is necessary to pass before arriving at the solution. It will also be necessary to define which agent will be in charge of passing from one state  $e_i$  to  $e_j$ , and the agents will have to coordinate in order to fulfill the global plan. Additionally, the norms that indicate the method of communications among the agents (obligations, permission, etc.) will have to be defined in the architecture, as well as the services that each agent can perform and with which role.

According to the model put forth in this study, at a given time  $t$  during the life of an organization, there are three possible situations that can occur:

1. If a planning agent has a problem, it must prepare a replanning process on its own. If that is not possible (it cannot carry out the tasks assigned by the coordinator), it will be necessary to proceed to the reassignment of global tasks, which can alter the tasks assigned to each planning agent.
2. If the structure of the society changes because, for example, a new agent enters and forms part of the society, it would be necessary to globally reassign tasks so that all agents are replanned.
3. The same thing occurs if there is a change in the norms. It would be necessary to reassign tasks at the organizational level.

These situations will force the system to adapt itself dynamically to the initially established plan, adjusting the tasks that the organization must carry out accordingly. The aim is to minimize as much as possible any changes to the initial objectives. To this end, it is necessary to provide an individual plan for each agent. This should be simple enough since the plan that was initially chosen by each planning agent is the most optimal plan, the most replannable in case of a problem. However, if this is not possible, there will then be a global reassignment of tasks. The following section presents a case study that tests the defined model. To do so, it is necessary to have an agent organization

specialized in certain tasks (those related to tourism) and a plan that should be carried out by the agents in the organization. It will be necessary to apply the defined model to carry out the plan and to verify that it is possible to replan in the event of a mishap during the execution time.

---

## 8.4 CASE STUDY

---

In order to evaluate the model proposed in section 3 of this chapter, several experiments were carried out using the THOMAS architecture, a technology developing virtual agent organizations in open environments [Giret et al., 2009][Carrascosa et al., 2009]. The reorganization and adaptation features in the agent's behavior are necessary for this platform, for which we have proposed the model outlined in this study, which will be tested in a case study. The objectives of the case study are:

- ❖ Develop an agent society that includes:
  - Definition of agent roles: we will define the general features required for the correct functioning of the society, as well as those of the defined case study. This will determine the behavior of the agents that comprise the organization. It is important to note that we will use an open system, which means that the definition of these roles should not make it impossible to carry out other similar case studies.
  - Definition of the necessary services for the correct functioning of an organization.
  - Definition of the norms that control the society and influence the interaction and behavior of the agents.
  - Definition of the messages and interactions of the agents themselves.
- ❖ Simulation of the behavior of the organization in a specific case involving the coordination and adaptation of its agents.
- ❖ Validation of the proposed planning model by simulating the organization in a case study.

This section is divided into 3 parts. The first will describe the context in which the case study was developed, presenting both the internal system features (general functioning of the model) and the external (agents that exist in the world). The second part will provide a brief explanation of the concept of an intelligent virtual agent, used in the case study simulation. Lastly, section 8.4.3 will present the system designed for this study.

---

### 8.4.1 Context

---

The case study presents a society whose aim is to offer tour guide services to tourists. This society is implemented by using the model proposed in section 8.3 and is represented in a virtual world containing a set of cultural heritage sites. The virtual worlds enable the incorporation of numerous interactive capabilities between the system elements, making it possible to carry out a significant number of tasks. The simulation within the virtual world represents a tourist environment in which there are guides and tourists, and in which the tour guide's tasks will be performed in adherence to a defined set of norms.

To establish the interactive model, it is necessary to analyze the needs and expectations of the potential users of the proposed virtual world. From this analysis it is possible to deduce the roles of the users that participate in the virtual world, and the manner in which they exchange information. The roles that have been identified within the case study are:

- ❖ *Tourist*: This agent is in constant communication with the Coordinator agent to obtain the planning routes it requests within a defined area, and to know which tour group it is assigned to. An area will be a location within the virtual world with certain characteristics that may be appealing to the tourist. For example, within a given area there can be several museums, historical buildings, scenic routes, musical events, etc. The profile for the tourist agent will be characterized by a set of variables specific to the agent (time, available funds, preferences, etc.).
- ❖ *Monument*: A monument agent can be one of two types: static (for example, a museum or building), which is characterized by its immobile state within our world; or dynamic (for example an exhibit or concert), which may or may not take place within our world at a given time. The attributes for each type are similar.
- ❖ *Guide*: An agent guide will be a specialized Tour agent with additional capabilities. In this case, besides being in constant communication with the Coordinator agent to obtain planned routes within the area, it will have a number of visitors assigned to it ("a group of tourists") to carry out the planned route. Included in its profiles are attributes such as the company it works for, the assigned group of tourists and visitors, the route to follow, the number of additional guides available, its specialized knowledge of certain monuments (preferred relationship with specific Monument agents), as well as its own abilities such as knowledge of a particular language. Within the society, the Coordinator agent will be in charge of assigning route, tasks and visitors to the available guide.
- ❖ *Visitor*: A Visitor agent is also a type of specialized Tourist agent, but with fewer capabilities. A Visitor will not have access capabilities through mobile devices, but will have its own profile with time, funds and preferences, and will be assigned to a Guide agent to follow a route within the area. These agents are a static specialization of the Tourist agent (their data do not change much over time).

- ❖ *Coordinator*: An agent that assumes this role will be able to assign the routes to take for both the Guide agent and the Tourist agent. It also has reorganization capabilities.
- ❖ *Notification*: This agent will be in charge of sending information throughout the system.
- ❖ *Manager*: This agent is responsible for the overall control of the system. It controls the connection and disconnection of the Tourist, Visitor and Guide agents to the system.

A Guide agent can automatically and dynamically generate the routes that a tourist or guide will follow, independent of other agents, but depending on the resources and profile for each of them. It will be the planning agent of the proposed organization. The Coordinator agent will be responsible for assigning each agent with the tasks to carry out according to the best global plan. In other words, there may be certain individual restrictions for each agent that must be taken into consideration by the Guide agents in order to generate a route, but it is the Coordinator agent who takes into account the restrictions for the entire society of agents in order to distribute and assign tasks to the Guide and Tourist agents. We can say that the Guide agent as well as the Coordinator agent will have access to the information on monuments, guide and tourist profiles, etc., stored in the system.

The information stored is a data base of cases that considers the task performed and the resources generated by each agent. When the Coordinator agent should assign a task, it must take into account the case base of all the tasks and agents, and retrieve those that contain at least one of the tasks that are proposed within the objectives of the organization. This way the Coordinator agent ensures that the task can at least be carried out. If none of the tasks are found in the case base, it would be necessary to look at the success probability of each agent to carry out the given task.

Figura 12 provides an outline of how the system functions. First of all, we can see the CBR system that coordinates the organizational tasks and stores the tasks carried out by the agents. We can also see how each CBP-BCI Guide agent, as suggested in [Corchado et al. , 2008], is assigned a cycle in which it will obtain its personalized optimal plan.

This is not a typical guiding system in which each guide is assigned to a group of tourists and visitors, and follows the tourist route for the group as assigned by the business office. Our case study uses a much more open system. The initial planning, both in terms of assigning tourist (and visitor) groups to a guide, and in determining the routes to take, can vary. A dynamic plan is created for a specific moment, according to the guides available, the preferences and location of the tourists and the events available on this particular tour. This may force the agent society to adapt itself to new requirements.



For example, if at a particular moment the tour stops for a “museum visit”, it may be necessary to form two groups of 5 tourists to enter the museum. If the groups were initially composed of 10 individuals, the system should be able to reassign the groups (change the structure of the organization) and introduce a new agent that can perform the role of Guide for the newly formed group.

As another example, if there is a change in the tourists’ preference, it may be necessary to change the group of tourists (and the guide assigned to them) at a particular time so that they may visit other parts of the world that are more in line with their interests (exhibits of a particular nature, shorter visits to monuments, etc.).

---

#### 8.4.2 *Intelligent virtual agents*

---

To simulate agent behavior, we will use the IVA (intelligent virtual agent) environment [Aylett & Luck, 2000] [Lozano & Calderón, 2006] in which intelligent agents are represented in three-dimensional models and form part of a simulated virtual environment [West & Hubbard, 1998].

An intelligent agent is a software entity that draws from its own knowledge of the environment to perform operations, whether on its own initiative or because a particular situation calls for it, that fulfill an objective. Intelligent agents can be considered individual entities within a program, as in our virtual environment, that control their own life and movement within the same environment. An intelligent virtual agent possesses characteristics that are exclusive to virtual environments [Aylett & Luck, 2000] and intelligent agents:

- ❖ They reside within a 3D simulated environment.
- ❖ They are represented as a 3D graphic within the world they live in and are capable of perceiving, adapting and reacting to their environment.
- ❖ They can express their behaviors graphically, as would a live being.
- ❖ Although they only exist and function within a specific environment, they are aware of the changes that are produced around them and are capable of responding to them autonomously.

The goal of the behavior (animation) associated with these agents is to make it as realistic as possible. There are AI techniques that are useful in reaching this goal [Caicedo, 2000] [Becheiraz & Thalmann, 1998]. The complex behaviors are particularly important in this type of intelligent virtual environment, and can be divided into the following categories:

- ❖ Motivated by emotions that are felt by the actor as a result of perceiving objects, actions and/or events. These are characterized by conditions for each emotion and variables that measure their intensity.
- ❖ Motivated by beliefs or reasoning, whereby agent behavior is oriented towards an objective. The decision-making process includes an

objectives planner, dynamically managing beliefs, and evolving objectives, internal states and levels of trust.

The present study focuses on the second category. We have seen how CBP-BDI agents provide the integration of reasoning and planning mechanisms that enable carrying out functions of adaptation and reorganization. We have also examined the principle techniques that try to solve the problem of finding a sequence of transitional states from a starting point (state) to a final destination (state), locating the optimal solution that solves the problem according to the target goal. Lastly, we have defined a social model to perform adaptive planning within an agent organization. Now we take all that has been set forth in this study and apply it to a real case study.

---

#### 8.4.3 *Multi-agent environment for our planning model*

---

*TouristOrg* is a multi-agent organization-based system. The system was designed according to the findings in [Agüero et al., 2009a][Agüero et al., 2009b][Agüero et al., 2010], which apply a MDD (*Model Driven Development*) focus on organization-oriented methodologies. It is possible to design an organization that is unified, intuitive, and has a high level of abstraction. Given these features, it becomes easier and simpler to design a MAS for organizations and obtain a model for a virtual organization that can be implemented on different platforms. The fundamental idea is to create different models for different levels of abstraction, and then combine them to achieve a full implementation. We used GORMAS [Argente, 2008] as the design methodology and THOMAS [Carrascosa et al., 2009][Giret et al., 2009] as the final platform design.

GORMAS (*Guidelines for Organization-based MultiAgent Systems*) is a guide methodology for the design of open MAS from the perspective of human organizations. It includes an analysis phase, a structural organization design phase, and a dynamic organization design phase. Together these phases specify which services the organization will offer, what the internal structure is and what norms control its behavior. Following the guidelines indicated in the methodological guide from [Argente, 2008], one of the first tasks is to instantiate a functional view (mission) of the organizational model, which represents the products and services offered by the system, the type of environment, the global objectives that is pursuing (mission and justification), the interest groups that are affected, and the information they use. We can see this in Figura 14.

The main service or product that the system offers is the tourist routes, which are required by the guides and the tourist themselves. Another product it offers is the cultural information about the monuments in the system. The service responsible for providing information on the characteristics of each monument is the Monument agent. The Guide agents can also be considered a “product”

since they can contain personal information in their profile, which they generate themselves, although the same information can be used and consulted by other members of the organization that must perform their assigned tasks. In this case, a guide could: (i) be a client when it wishes to obtain system information about a particular route to follow, or about the monuments it has been assigned to; (ii) be a provider when it offers route plans generated by the system or information about its own profile "Service Information Guides". The system will offer the necessary services for generating and assigning routes to the guides, it will consult the information for each guide (profile), assign guides to the tourists, and consult information on the monuments (cultural information). It can also solve incidents (for example, whether a monument is or is not available) and generate new routes. These incidents are provided by the Manager, which is in charge of informing on everything that occurs in and for the system. The mission of the organization is twofold: to minimize the guiding system while attempting to generate the best routes with the fewest resources (time, money, etc.); and to maximize production, i.e., to generate more benefits by creating as many routes as possible. The two objectives are complementary and proportional, meaning that a minimal cost with the most routes implies the greatest production.

Each of the services identified in the previous functional view is detailed and a corresponding activity model is created. The functions offered are also established; this includes the services that should be publicized as well as the policies involved in acquiring or releasing roles. Decisions are made regarding which functions should be implemented by internal agents in the system, and which will be provided by external agents. This makes it possible to model the dynamic nature of the organization as an open system. The Monument, Tourist, Visitor and Guide roles are accessible to external agents, but must be acquired through the proper procedures. The Manager, Notification and Coordinator roles are associated with internal agents, as they are not accessible. Figure 3b shows the organization model and indicates the external agents (A-Agents), the internal agents (Agents) and the roles that each assumes (Figure 3b).

In our case, the management and services roles associated to the organizational units will be made available through the THOMAS platform [Carrascosa et al., 2009] [Giret et al., 2009]. An OMS (Organization Management System) offers the necessary services for the correct functioning of an organization. It provides the services needed to register or unregister structural components, specifically the roles, norms and units that already exist in the system. It also offers services that identify each component.

The virtual environment in which our agents would reside will include the stored documentation of tourist information based on images, phrases and other contents. For our simulation, we selected 20 virtual agents that represent visitors, an average of 7 guides, 1 dynamic monument and 3 that are static. For the simulation model we used REPAST [North et al., 2007][REPAST2009], a

collection of Java libraries that allow the developers to create simulated environments and agents in social networks, to gather simulation data automatically, and to easily build user interfaces. Figura 18 shows 3 of the 3D models used to represent the virtual environment to the three Monument agents identified in the case study.

The simulations were performed with Visitor agents (not Tourist agents) since their data does not change throughout the testing period. To evaluate the adaption of the model for replanning and reorganizing, we used variations of the Guide agents.

The agents that take on the role of Guide are those that will carry out dynamic planning according to the tasks they need to carry out for each group of tourists. The generated plans should ensure that all of the visitors assigned to a tour guide are able to follow their tourist route. They will be personalized according to the Guide's profile and work habits, and should take into account the restrictions directly related to each agent on an individual basis, as well as the restrictions of the organization itself. These restrictions are imposed according to the norms for the society of agents. In the proposed model, we could see, for example, that at a specific time during the life of the organization, there were three kinds of situations that could require a dynamic replanning: (i) the agent could have a problem, (ii) the agent could change the structure, or (iii) the norms for the society could be modified. In our case study, the norms for the society included:

- ❖ the work schedule for a Guide agent (8 hours);
- ❖ the maximum number of Tourist agents assigned to a guide;
- ❖ visiting days and hours for certain monuments;
- ❖ the maximum number of Guide agents that can participate on a route;
- ❖ the minimum number of points to visit on a route.

In light of possible changes that an agent may make while executing a task, the society should be capable of reorganizing dynamically, of adapting to the changes and continuing with the plans that enable them to reach the initially stated objectives. In a previously mentioned example, it was necessary to form groups of 5 tourists in order to enter a certain museum, although the initial groups were made up of 10 individuals. This is the type of situation in which the system should be able to reassign groups and introduce a new agent that can perform the role of Guide for the newly formed group. This case produces a global reassignment. An example of an individual replanning situation within the case study may involve a delay in entering a museum (in a live situation, there could be long lines, for example). In this situation, the Guide agent performing the task can replan an individual plan and leave the task of visiting the museum for another time. This would produce a re-ordering of its tasks. If this were not possible, however, it would be necessary to reorganize the remaining Guide agents, which would call for a global reassignment.

While the system adapts to the changes, taking the planning model into account, it will at all times attempt to minimize costs related to the objective; costs that are proportional to the time that a Guide agent is working, the resources used, and the associated profile. The profile of a Guide agent is related to its specialization. The known information for each agent includes:

- ❖ type of monument it specializes in (painting, sculpture, etc.);
- ❖ language(s);
- ❖ size of group it can handle;
- ❖ type of group it specializes in (children, seniors, etc.).

These different options were taken into account for the case study, but it would be possible to add more features according to the desired level of detail that we want for the Guide. For example, if a Guide specialized in a certain type of monument is assigned to a route in which it is not specialized, it would have a greater associated cost than would another Guide that is specialized in the same route.

If it were necessary to recover from a system failure, the monitoring process would still continue. Each agent stores its own data within a data base. Because they are most prone to crashing, the Tourist and Guide agents save their data every hour. When an agent crashes, THOMAS uses the most recent back up file to generate a new instance of the agent. The database and the server should be able to recover from crashes and control redundancies. To this end, the system chosen for the server was the RAID (Redundant Array of Inexpressive Disk) system. The virtual world simulation is executed in the server, which works in collaboration with the THOMAS architecture (Figura 19)

---

## 8.5 EXAMPLES OF PLANNING

---

Guide agents can be considered the planning agents of the organization. The objectives of the Guide agents are to dynamically generate the plans that they must carry out, and to maintain reports on the tasks they perform. This is done to ensure that the Visitors assigned to the Guide are able to follow their route. In other words, the Guide agent plans the workday by obtaining plans dynamically according to the necessary tasks requested by the Visitors. The Guide agent offers 3 basic services within the architecture:

- ❖ *ConsultGuide*: to consult data related to a specific Guide,
- ❖ *SolveIncident*: to solve situations (such as if a monument is closed, or a tourist must abandon the route) and replan if necessary,
- ❖ *GenerateRoute*: to generate the most replannable solution (plan) for the agent organization, given a list of preferences for a particular problem.

If there is a set of beliefs  $B$  compatible with the planning problem  $E$  that the agent organization wants to carry out, it is possible to generate a plan that contains all of the possible plans produced by the combination of compatible beliefs.

One of the tasks is a Java objective that contains the data for a Visitor or Tourist agent that requests a service, a description of the service, and the time constraints in carrying it out. One or more goals are established for each task. A goal is also a Java objective, and identifies what a Guide agent wants to achieve (completed task) and under what restrictions. To achieve its goals, the agent generates plans that are also processes written in Java code. One goal can create new goals (sub-goals). For example, for the task "Visit Museum 1", the Guide agent creates a new goal for each specific execution. The CBP system builds plans as a sequence of tasks that the Guides need to carry out. A description of a problem will be formed by the tasks that each Guide agent needs to perform, the available resources, and the allotted time. An example of task can be seen in Tabla 3

For every task  $j$ , the Coordinator agent is responsible for assigning a Guide  $i$  to carry out that task. Prior to assigning the task, the Coordinator agent must take into consideration the case base in which each Guide agent stores the cost associated with performing that task in the past. Once the Coordinator has identified all of the agents in the organization that are needed to carry out the plan, it assigns each task to the agent responsible for completing it. At that moment each Guide agent becomes aware of its tasks and designs an individual plan. Each Guide agent is a type of CBP-BDI agent capable of providing efficient plans in execution time. To do so, it incorporates a case-based planning (CBP) planner. The purpose of these agents is to adapt solutions that have been used to solve problems in the past. In this case, the new problems are defined by the set of tasks that have been assigned by the Coordinator agent. Knowing the tasks it must complete, the Guide agent goes through its cycle to generate the most replannable plan (MRPI) in execution time [Corchado et al., 2008].

If at any time during the execution one of the Guide agents is unable to follow through with the plan it had created, it will attempt to replan the task by generating a new plan with the tasks assigned by the Coordinator agent. If it is unable to obtain a new plan, it will have to inform the Coordinator agent, who can then reassign tasks among the remaining Guide agents in the organization. In this case, the Coordinator agent will have to take into account the tasks that have been performed up until that moment, and generate a new global plan in which each Guide agent is reassigned with some of the tasks that still need to be done. Figure 2 illustrates this process. The organization is capable of adapting with minimal change to its structure. It will attempt to replan the tasks at the agent level; if this not possible, it will implement a reorganization in which all tasks are reassigned among the agents within the society. As a final option, it will modify its structure (for example, bringing new agents into the

organization, or reorganizing the groups of agents). We will now look at this process in greater detail.

With regards to generating plans for each Guide agent, it is important to underscore that during the retrieve phase of the CBP cycle, the agent retrieves the most similar descriptions by applying various similarity algorithms (cosine, clustering, etc.). During this phase, it is possible to search the beliefs base to recover descriptions of problems within a range of similarity closest to the original problem. In our case we allowed a 20% tolerance level. Once the majority of the descriptions have been retrieved, we can obtain the solutions associated to them. A solution contains plans (sequence of tasks) that have been carried out in the past (assuming that replanning is possible) to solve similar problems focused on meeting the objectives that a tourist Guide has on a given work day. The efficiency rate is also calculated for every solution.

A Guide agent can control the execution of a plan. It takes into consideration the objectives and resources needed for each task, as well as the objectives that it must achieve and the resources available for carrying out its plan. The objectives or plans that each Guide has are determined according to the previously mentioned restrictions, such as workload (8 hours) or the time available for a given problem (time for visiting a tourist site). The available time is a problem restriction that will also influence the hyperplane of restriction, specifically the ranges of positive values on the z-axis. The resources needed for the tasks can include entry tickets for a museum, money, or equipment needed to visit a particular tourist site. With this information the Guide agent can handle the incidents and the interruptions that can occur during planning.

For each Guide agent in the organization, it is necessary to complete the process of assigning tasks. We have seen how the Guide agents individually calculate the optimal plan for the tasks they must perform. Prior to this, however, they must know which of all of the planning tasks in the organizations are the ones that only they must perform. It is necessary to implement a global reassignment in the agent organization so that tasks are distributed as optimally as possible among all of the Guides in the organization. This is the job of the Coordinator agent. The following example shows how the Coordinator agent works.

First of all, it is necessary to bear in mind that each Guide agent has a different profile according to its qualifications and the tasks it tends to carry out. Given a state “j” for each agent “i” where  $i \in \{1, \dots, m\}_{m \in \mathbb{N}}$ , we can define the tuple  $z_{ij}$ , where each coordinate of the tuple refers to a characteristic that defines it. In other words we consider the vector in which the profile of each Guide agent is stored in the virtual world. It is appropriate to manage the Guide profiles since our intention is to maximize efficiency, for which we must minimize costs, and these depend on the profile of each Guide agent. When the tasks need to be



assigned to the Guide agents, all of the previous experiences as well as the profile of the Guide are retrieved and taken into account for the new problem. The tasks correspond to the number of Tourists and/or Visitors (a group) that will be assigned to one of the Guides in the organization. The profile of a Guide includes the specializations of that agent. For example, not all Guides can speak all languages. If one Guide is better qualified to speak a particular language, it will be assigned to the group that speaks that same language, although other restrictions must also be accounted for, including workload (8 hours). Consequently, the number of tourists and visitors assigned to a Guide depends on the time required to complete a tourist route. The Manager agent will consider the level of satisfaction that the tourists indicate at the end of the route, as well as the arrival of new tourists to the organization .

Secondly, the time needed for each task must be stored in the memory of beliefs. The time is described as  $t_{ij}$ , the time that Guide agent “i” takes to complete the state or tasks “j” where  $t_{ij} = \text{Máx}_k \{ t_{ijk} \}$  and  $t_{ijk}$  indicates the time that agent “i” takes to perform task “j” for tourist “k”. Taking the maximum value of “k” (type of tourist), we can ensure that the guide has enough time to complete the tasks, regardless of the type of tourist. These times are initially estimated by simulating the visits prior to offering the service to the tourists.

Of all the available Guides, priority is given to the Guide whose profile is most compatible (since it is only possible to assign certain types of tasks to the Guide agents that can actually carry them out).

Once the process of assigning tasks to the Guide agents is complete, the assignment is communicated to each Guide. At that point, the planning process begins. The Guide agent should keep in mind the time it has available, and the time required for each task that it has been assigned. Additionally, it must consider the necessary and available resources. In the process of creating a new plan, the Guide agent recovers the cases from the memory of beliefs that contain a problem description similar to the current problem, and then combines the solutions (plans) that were used to solve them. A number of different measures are involved in normalizing the time needed to arrive at a given museum, or to take a group of tourists or visitors from one museum to another. These times are included directly in the time assigned for each task.

A plan can be interrupted for a variety of reasons. Those that have been implemented into the virtual world include: lack of resources (insufficient funds to purchase tickets), unavailability of a Guide and the need to assign a new one (global reassignment), inability to complete a visit within the allotted time (individual replanning), an emergency situation, etc. If a Guide agent is in a situation involving a plan that is interrupted, it rejects the original plan and



proposes an alternative. The first action that the agent will attempt is to replan the tasks while attempting to maintain the original assignments made by the Coordinator agent. The new plan should meet the original objectives and restrictions. If this is not possible, it will be necessary to reassign tasks to the Guides. This reassignment is done while limiting the number of required changes, while at once taking into account the tasks that were originally assigned to the Guides, the development of the plans (the tasks that have been performed and those still pending) and the Guide profiles (prioritizing the tasks that were not completed). When a new task is introduced, there is a reassignment and the Guide who receives the task must replan in order to include the new task in its plan.

Lastly, depending on the plan's efficiency, it will be stored according to its level of efficiency in the beliefs base so that it can be used for similar problems in the future. Similarly, the tasks performed by the Guide agent will also be stored for future use by the Coordinator agent to carry out the next task assignment for the organization. The following paragraph provides a detailed example.

Let  $E_g = \{e_0^g, \dots, e_h^g\}$  be the tasks carried out by a group of tourists and visitors "g" in order of priority. We have the following problem  $E = \bigcup_g E_g = \{e_0, \dots, e_n\}$ , where E represents the complete set of tasks that must be completed (for this reason they are not superscripted). Let us assume there are 10 guides. Randomly selecting a Guide  $i \in \{1, \dots, 10\}$ , (specifically,  $i=3$ ), the task assignment according to their profile is:

1. Agent Task: Visit the cathedral with tourist group 2  $\equiv e_1^2$ ;  $t_{31}=30$  min.
2. Agent Task: Take tourist group 2 to the aqueduct  $\equiv e_2^2$ ;  $t_{32}=15$  min.
3. Agent Task: Take tourist group 2 to the hermitage  $\equiv e_3^2$ ;  $t_{33}=10$  min.
4. Agent Task: Visit the hermitage  $\equiv e_4^2$ ;  $t_{34}=10$  min.
5. Agent Task: Take tourist group 2 to the Roman city  $\equiv e_5^2$ ;  $t_{35}=20$  min.
6. Agent Task: Visit the Roman city  $\equiv e_6^2$ ;  $t_{36}=30$  min.
7. Agent Task: Take tourist group 2 to the ravine  $\equiv e_7^2$ ;  $t_{37}=50$  min.
8. Agent Task: Hike along the ravine with group 2  $\equiv e_8^2$ ;  $t_{38}=20$  min.
9. Agent Task: Return to the cathedral with group 2  $\equiv e_9^2$ ;  $t_{39}=10$  min.

Calculating the assigned tasks ensures both that the total amount of time assigned to a Guide does not exceed 8 hours, and that any other restrictions corresponding to the norms of the organization are also respected. Once the

task assignment is complete, the Guide agent implements the generation and execution of its plan. To do, it searches the beliefs base to retrieve past assignments with similar characteristics, as well as the corresponding plans that were used to carry out those assignments, and then generates a new plan. To execute the new plan, the Guide agent displays each task on a mobile device. Once the task is completed, the Guide agent asks a human guide to input the results obtained after having completed the task. Each task has a set of objectives that must be met so that the global plan can be successfully completed. To perform each task, the Guide agent should have the number of available resources. For example, the task "Buy tickets for museum 1" corresponds to the objective "Visit museum 1"  $\equiv O_0$  and "breakfast, lunch, tea and dinner" correspond to the objective  $\equiv O_{2,4,6,7}$  (task 2 indicates breakfast, task 4 indicates lunch, task 6 indicates tea, and 7 indicates dinner). A similar coding is used for resources. It was decided that the objectives and resource variables should be dichotomic (binary) with a value of 0 to 1 in order to indicate the absence or presence of a resource or objective. As shown in Figura 20a, value 1 indicates the resource that is needed or the objective to be met, while zero denotes the contrary. Figura 20a shows the representation of a space  $\mathfrak{N}^3$  for tasks according to the following three coordinates: time, number of objectives achieved, and number of resources used (coordinates taken from similar retrieved cases). Specifically, Figura 20a shows a hyper plan of restrictions and the plan followed for a case retrieved from the beliefs base, considered to be similar to the new case. The retrieved case includes a group of tourists that want to visit the Cathedral museum, the aqueduct, the hermitage, the Roman arch and its surroundings, and the ravine. The distance between the Cathedral and the aqueduct can be covered on foot, but the rest must be traveled by bus due to the geographic location of each point of interest, the visiting hours, etc.

- ❖ The cathedral has a schedule of liturgical events and visiting hours for the museum.
- ❖ The hermitage does not have a museum, but does have liturgical events and is surrounded by beautiful landscapes.
- ❖ The aqueduct, the arch and the ravine must be seen in the light of day.

There are 120 possible routes, not all of which are viable because of the previously mentioned restrictions. In a simulated scenario where the Coordinator assigned this group of tourists to a Guide, the planning process used by the Guide for the tasks it needed to perform is the same as that shown in Figura 20a.

The guide met with the group at the entrance to the Cathedral, provided a guided tour of the Cathedral (30 minutes), then continued on foot to the aqueduct (15 minutes). From there the group rode on the bus (10 minutes) to

visit the hermitage (10 minutes), got back on the bus (20 minutes) and went towards the ruins of the Roman city (30 minutes visit). Once again the tourists rode on the bus (50 minutes) to arrive and hike along the ravine (20 minutes). This route is the most replannable because of the location of the hermitage. While boarding the bus at the aqueduct, we were informed that there was a wedding at the hermitage and that the bride arrived late. So as to not arrive during the ceremony, the route is replanned, resulting in the plan shown in Figura 20b.

The group rides the bus (15 minutes) towards the Roman ruins (30 minute visit), then boards the bus again (50 minutes) to arrive at the ravine for a hike (20 minutes). From there the bus (20 minutes) heads towards the hermitage (10 minutes), the tourists board the bus once again (10 minutes) and arrive back at the entrance to the cathedral. The task assignment for Guide 3 would be as follows:

1. Agent Task: Visit the cathedral with tourist group 2  $\equiv e_1^2$ ;  $t_{31}=30$  min. Done.
2. Agent Task: Take tourist group 2 to the aqueduct  $\equiv e_2^2$ ;  $t_{32}=15$  min. Done.
3. Agent Task: Take tourist group 2 to the Roman city  $\equiv e_3^2$ ;  $t_{35}=15$  min.
4. Agent Task: Visit the Roman city  $\equiv e_6^2$ ;  $t_{36}=30$  min.
5. Agent Task: Take tourist group 2 to the ravine  $\equiv e_7^2$ ;  $t_{37}=50$  min.
6. Agent Task: Hike along the ravine with group 2  $\equiv e_8^2$ ;  $t_{38}=20$  min.
7. Agent Task: Take tourist group 2 to the hermitage  $\equiv e_3^2$ ;  $t_{33}=20$  min.
8. Agent Task: Visit the hermitage  $\equiv e_4^2$ ;  $t_{34}=10$  min.
9. Agent Task: Return to the cathedral with group 2  $\equiv e_9^2$ ;  $t_{39}=10$  min.

To keep Figura 20a from being too large, and in order to easily appreciate the plan, the time axis has been rescaled (axis  $z$ ), establishing an isomorphism between the intervals  $[0, 1]$  and  $[0, 8]$ :

The isomorphism is as follows:

$$\begin{aligned} \lambda : [0,1] &\rightarrow [0,8] \\ z &\rightarrow \lambda(z) = 8 \cdot z \end{aligned} \quad (8.17)$$

For similar cases that are retrieved, the procedure is the same. The new plan for the agent is generated in parts so that it is surrounded by the greatest density of plans.

The figures illustrate the plan as it was carried out. To understand the graphical representation, let us focus on the initial task  $e_1$  and the final task  $e_9$ . In between these two tasks, the Guide agent could carry out other tasks that would involve the same or different tourists and visitors.

The idea presented in the planning model is to select the optimal plan, the one with the most plans surrounding it, as the solution. We can see in Figura 20b that the selected plan is the one with the most plans surrounding it, having tasks that could be reassigned if the plan were interrupted.

Another plan proposed for our example was:

- ❖ Organization task: Assign Guide 1 to group 2, Guide 2 to group 4, Guide 3 to group 5, Guide 4 to group 6
- ❖ Agent Task: Take tourist group 2 to the cathedral  $\equiv e_1^2$ ;  $t_{11}=30$  min.
- ❖ Agent Task: Visit the cathedral with tourist group 2  $\equiv e_3^2$ ;  $t_{13}=60$  min.
- ❖ Agent Task: Serve breakfast to tourist group 2  $\equiv e_2^2$ ,  $t_{12}=60$  min.
- ❖ Agent Task: Serve breakfast to groups 4, 5 and 6  $\equiv e_2^4, e_2^5, e_2^6, t_{v5}$   
=120 min with  $v \in \{2, 3, 4\}$
- ❖ Agent Task: Hike along the ravine with 2,4,5 and 6,  $\equiv e_4^2, e_4^4, e_4^5, e_4^6$ ,  
 $t_{s4}=120$  min with  $s \in \{1, 2, 3, 4\}$

During the visit to the cathedral, it is necessary to reorganize the structure of the system since the limit is four people per group. A new guide must appear in the plan since group 2 has 6 people. The profile of the new guide must also include the ability to speak French since it is the language spoken by the tourists in the group. The new plan would be as follows:

- ❖ Organization task: Assign Guide 1 to group 2
- ❖ Agent Task: Take tourist group 2 to the cathedral  $\equiv e_1^2$ ;  $t_{11}=30$  min.
- ❖ Organization task: Form new Visitors Unit: group 7
- ❖ Organization task: Assign Guide 5 to group 7
- ❖ Agent Task: Visit cathedral with tourist groups 2 and 7  $\equiv e_3^2, e_3^7$ ;  $t_{b3}$   
=60 min with  $b \in \{1, 5\}$
- ❖ Agent Task: Serve breakfast to tourist group 2  $\equiv e_2^2$ ,  $t_{12}=60$  min.
- ❖ Agent Task: Serve breakfast to tourist groups 4, 5 and 6  $\equiv e_2^4, e_2^5, e_2^6$ ,  
 $t_{v5}=120$  min with  $v \in \{2, 3, 4\}$

- ❖ Agent Task: Take groups 2,4,5 and 6 on a hike along the ravine  $\equiv e_4^2$ ,  
 $e_4^4, e_4^5, e_4^6, t_{s4} = 120$  min with  $s \in \{1, 2, 3, 4\}$

In this example, the organization can adapt the time according to the changes that affect the member agents. The Guide agents can plan tasks as well as replan tasks, and the Coordinator agent can reorganize and redistribute tasks among all agents if they are themselves unable to adapt to the change.

The mathematical calculations for obtaining  $h(x)$ , using Duchon techniques, the set of geodesics  $\{gn(x)\}_{n \in \mathbb{N}}$  960 using the resolution of the Euler and transversability equations, or for obtaining the Jacobi field, are carried out with the @Mathematica 5.1 program and the Jspline+ and Jlink java libraries.

---

## 8.6 RESULTS AND CONCLUSIONS

---

There is no doubt that the results of this study are an important aspect of the case study designed for the organization. However, the objective of the study is not to provide a precise simulation of the behavior that takes place within a touristic environment, but to validate the proposed planning model through the organization simulation for the case study. In order to interpret the results most conveniently, it is necessary to carefully study the functioning of the proposed scenario to determine whether the planner works. The following studies were carried out:

Given the same tourists attractions to be visited on the same day, and the same number of tourists per group, one group used the planner and the other did not.

The results for different days, as far as the number of Guide agents used, can be observed in Figura 22a. The color blue represents the average number of guides needed each day using the planner, and red the number without using it. The proposed model helps the organization utilize fewer guides, thus minimizing its costs.

Another important point for evaluating the effectiveness of the planner is to compare the efficiency between when it was and was not used. We can recall that efficiency relates the cost of using the agents with the number of sites visited. Using the previous case in which there was a specific number of tourist sites to visit, the efficiency for both cases can be represented graphically (Figura 22b).

The results of the study establish the model, along with the architecture presented in this thesis (THOMAS), as a powerful alternative with respect to other architectures used for developing agent organizations. The restrictions related to the use of the planning model are minimal. Additionally, the diversity

of elements that influence how the proposed plans are generated makes it easily adaptable to diverse contexts and problems.

The proposed social planning model is among the very few that offer the possibility of deliberative and social behavior. It is worth mentioning that this is a unique model that incorporates its own reorganization and social adaptation mechanism. THOMAS facilitates the development of MAS in an organizational paradigm and the social model adds the reorganization and adaptation functions.

Assigning and reorganizing tasks in the proposed planning mechanism could have been carried out in a variety of different ways, three of which are shown in Figura 25. In order to evaluate the results, two of these possibilities were compared in order to validate how the system would react if instead of a distributed coordination, it would have used a model in which the planning of all the tasks in the organization was done in a centralized manner.

Using a centralized planning method to manage the organization, there is a leader agent (coordinator and planner) responsible for generating the plans that will be subsequently carried out by the remaining agents. To ensure that each plan is correct, the organization needs to acquire a great amount of information about the tourists, the guides and the monuments in order to determine whether the followers (guides) can correctly carry out their tasks.

The distributed planning diagram reduces the amount of required information by distributing its objectives. In this case, the coordinator will have to take into account a subset of information about the collective state, with less precise information about the guide agents, and produce an abstract plan with enough detail to be able to distribute tasks objectively among the guides. It sends specific objectives to each guide who in turn generates its optimal solution to carry it out.

Another way of coordinating this effort would be to use contract networks in which a leader announces a task to a group of followers, each of which proposes a way to carry it out, and the leader selects the method to use according to the best offer. This is unnecessary in our case because of the CBR mechanism that always assigns tasks optimally, taking previous experiences into account.

Comparing the first two models, the system showed significant improvement when increasing the number of tasks, reducing the amount of time used to plan and replan the tasks (Figura 26).

The MAS presented in this thesis uses THOMAS in a dynamic environment, as well as the self-adaptation capabilities that were added. THOMAS can be compared to other platforms currently available to create agent organizations. Each of them should take certain aspects into account, such as structural

organization (supporting groups, assuming roles, topology, interactions, and social rules) and the dynamic aspect of the organization (entry and exit of external agents, creating roles, life cycle of the agents, controlling behavior, adaption and coordination). Tabla 5 compares some of the most popular platforms and shows the available support or exclusion of these aspects. THOMAS provides a clear advantage with respect to the others. Additionally, along with the social planning model, it provides the reorganization and adaptation features required in dynamic environments, such as the one proposed in this study.

Perhaps the most similar model to the one proposed in this study is the model presented in the study on RETSINA (Reusable Environment for Task-Structured Intelligent Network Agents) [Giampapa and Sycara, 2002], in which proposed a social coordination model based on HTN (Hierarchical Task Network) planning and attempts to revise the plans that are carried out by the organization in execution time. Even so, in comparing the two proposals we can see how the model in the present study provides several advantages not included in the other. Our proposed model is capable of reorganizing the society in execution time and planning in execution time; but additionally it is capable of obtaining the optimal route for its organization, i.e., the most easily replannable route, and of learning from the experiences that occur throughout the life -cycle of the organization. This is due to the case-based reasoning mechanism included in the model. Along these lines, the research that is currently taking place involves the application of reasoning and learning techniques for agent organization models. Specifically, [Bou et al., 2008] [Bou et al., 2009] provides a proposal in which Automatic Electronic Institutions are combined with CBR to solve situations that have occurred in the past; however in this case, the aspect of dynamic adaptation and reorganization is not considered.

In conclusion, we can affirm to have achieved out stated objectives: (i) Develop agent societies; (ii) Simulate the behavior of an organization in a specific case involving the coordination and adaption of its agents; and (iii) Validate the proposed planning model through a simulation of the organization in a case study.

Lastly, the model used the THOMAS architecture in its very general state, without presenting particular restrictions to the domain during its application. Consequently, its ability to adapt to specific problems does not imply a diminished efficiency in the applications that are developed.





## BIBLIOGRAFÍA



---

## REFERENCIAS BIBLIOGRÁFICAS

---

- [Aamodt y Plaza, 1994] Aamodt A. y Plaza E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *Artificial Intelligence Communications*, vol 7 (1994): 1, 39-52.
- [Agent Oriented Software, 2004] Agent Oriented Software Pty. Ltd., *JACK Intelligent Agents - Agent Practicals*, 4.1 ed, 2004.
- [Agre y Chapman, 1990] Agre P. E. y Chapman D. What are plans for?. *Robotics & Autonomous Systems* 6, pp. 17-34. 1990.
- [Agüero et al., 2009a] Agüero, J., Rebollo, M., Carrascosa, C., Julián, V.: Developing virtual organizations using MDD. In: *Proc. Workshop on Agreement Technologies(WAT2009)*, pp. 130–141 (2009)
- [Agüero et al., 2009b] Agüero, J., Rebollo, M., Carrascosa, C., Julián, V.: Agent design using model driven development. In: *7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS2009)*, pp.60–69 (2009))
- [Agüero et al., 2010] Agüero J., Rebollo M., Carrascosa C. ,Julián V.: MDD for Virtual Organization design. In: *8th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS2010)* (2010)
- [Ali et al., 2008] R. Ali, V. Bryl, G. Cabri, M. Cossentino, F. Dalpiaz, P. Giorgini, A. Molesini, A. Omicini, M. Puviani, V. Seidita. 2008. MEnSA Project - Methodologies for the Engineering of complex Software systms: Agent-based approach, Tech. Rep. 1.2, UniTn.
- [Argente, 2008] Argente Villaplana, E.: *Gormas: Guías para el desarrollo de sistemas multiagente abiertos basados en organizaciones*. Ph.D. thesis, Universidad Politécnica de Valencia (2008)
- [Argente et al., 2004]E. Argente, A. Giret, S. Valero, V. Julian, V. Botti. 2004. Survey of MAS Methods and Platforms focusing on organizational concepts. In: Vitria, J, Radeva, p and Aguilo, I (ed) *Recent Advances in Artificial Intelligence Research and Development, Frontiers in Artificial Intelligence and Applications: 309–316*
- [Argente et al. 2006] E. Argente, V. Julian, and V. Botti. 2006. Multi-agent system development based on organizations. *Electronic Notes in Theoretical Computer Science*, 150:55–71.
- [Argente et al., 2008] E. Argente, N. Criado, V. Julian, and V. Botti. Designing Norms in Virtual Organizations. In *Proc. 11th Int. Workshop Artificial Intelligence of Catalan Association (CCIA08)*, page In Press, 2008.
- [Artikis, 2003] Artikis A. Executable Specification of Open Norm-Governed Computational Systems, Ph.D. thesis, Department of Electrical & Electronic Engineering, Imperial College London, November 2003.
- [Artikis et al., 2001] Artikis, A., Kamara, L., Pitt, J.: Towards an Open Agent Society Model and Animation, *Proc. of the 2nd. Agent-Based Simulation Workshop, Passau, 2001*, pp. 48 – 55.

## Referencias

---

- [Artikis et al. 2009] Alexander Artikis, Dimosthenis Kaponis, and Jeremy Pitt. Multi-Agent Systems: Semantics and Dynamics of Organisational Models, chapter in Dynamic Specifications of Norm-Governed Systems. IGI Global, 2009.
- [Axelrod, 1986] Robert Axelrod. "An Evolutionary Approach to Norms." *American Political Science Review* 80 (December): 1095-1111, 1986
- [Aylett y Luck, 2000] Aylett, R., Luck M.: Applying artificial intelligence to virtual reality: Intelligent virtual environments. *Applied Artificial Intelligence*, (2000)
- [Badler y Palmer, 1999] N. Badler, M. Palmer, T. Bindiganavale "Animation Control for real-time virtual humans", "Communications of the ACM", Agosto 1999, vol 42, num 8
- [Bajo et al., 2007] Bajo J., Corchado J.M. y Rodríguez S. Intelligent Guidance and Suggestions using Case-Based Planning. *Proceedings of ICCBR'07, LNAI*, vol 4626 pp. 389-403, Springer Verlag. (2007)
- [Bajo et al., 2006] Bajo, J., De Paz, Y., De Paz, J. F., Martín, Q., & Corchado, J. M. (2006). SMas: A Shopping Mall Multiagent Systems. *Proceedings of IDEAL'06, Lecture Notes in Artificial Intelligence (LNAI)*. 4224, pp. 1166-1173. Springer-Verlag.
- [Bajo et al. 2008] J. Bajo J, V. Julian, J.M. Corchado, C. Carrascosa, Y. de Paz, V. Botti, J.F. de Paz. An Execution Time Planner for the ARTIS Agent Architecture. *Engineering Applications of Artificial Intelligence* Volume: 21 Issue: 5 Pages: 769-784 Published: AUG 2008
- [Bajo et al., 2009] Bajo J., Corchado J.M., De Paz Y., De Paz J.F., Rodríguez S., Martín Q. and Abraham A. SHOMAS: Intelligent Guidance and Suggestions in Shopping Centres. *Applied Soft Computing*. Elsevier Science Ref. ISSN: 1568-4946, doi:10.1016/j.asoc.2008.11.009 . Vol 9, núm, 2, Pág: 851-862, 2009
- [Baumer et al., 2000] G. Baumer, M. Breugst, S. Choy, T. Magedanz. 2000. Grasshopper: A Universal Agent Platform based on OMG MASIF and FIPA standards, in: *Agents Technology in Europe*.
- [Becheiraz y Thalmann, 1998] Becheiraz, P., Thalmann, D.: "A Behavioral Animation System for Autonomous Actors Personified by Emotions" (1998)
- [Bellman, 1957] Bellman, R.E. (1957). *Dynamic Programming*. Princeton University Press, Princeton, New Jersey.
- [Bellman y Dreyfus, 2001] Bellman R.E. y Dreyfus S.E. *Applied Dynamic Programming*. Ed. Athena Scientific, Belmont, MA, 2001.
- [Bergmann y Stahl, 1998] Bergmann R. y Stahl A. Similarity measures for object-oriented case representations. In B. Smyth and P. Cunningham, editors, *Advances in Case-Based Reasoning*, volume 1488 of *Lecture Notes in Computer Science*, pages 25-36. Springer, 1998.
- [Bergmann et al., 2005] Bergmann R., Kolodner J. y Plaza E. Representation in case-based reasoning. *The Knowledge Engineering Review*, Vol. 00:0, 1-4.c 2005, Cambridge University Press
- [Berners-Lee et al., 2001] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 2001

- [Blum y Furst, 1997] Blum A. y Furst M. Fast Planning Through Planning Graph Analysis, *Artificial Intelligence*, 90:281--300 (1997).
- [Boella et al., 2005] G. Boella, J. Hulstijn, and L. Van Der Torre. 2005. Virtual organizations as normative multiagent systems," in *HICSS IEEE Computer Society*.
- [Boissier et al, 2006] O. Boissier, J. Padget, V. Dignum, G. Lindemann, E. Matson, S. Ossowski, J. Sichman, J. Vazquez-Salceda. 2006. *Coordination, Organizations, Institutions and Norms in Multi-Agent Systems*, Vol. 3913 of LNCS (LNAI), Springer-Verlag
- [Boissier, 2007] O. Boissier and B. Gateau. Normative multi-agent organizations: Modeling, support and control. In *Normative Multiagent Systems, 2007*
- [Bonet y Geffner, 1999] Bonet B. y Geffner H. HSP: Heuristic Search Planner. Entry at AIPS-98 Planning Competition, *AI Magazine Vol 21(2)*, 2000
- [Booth et al, 2004] D. Booth and et al. Web Services Architecture. W3C Working Group, Feb 2004. 11
- [Bou et al.,2007a] Eva Bou, Maite López-Sánchez, and Juan A. Rodríguez-Aguilar. Adaptation of autonomic electronic institutions through norms and institutional agents. To appear In *Engineering Societies in an Agents World*, Lecture Notes in Computer Science.Springer Verlag, 2007.
- [Bou et al.,2007b] Eva Bou, Maite López-Sánchez, and Juan A. Rodríguez-Aguilar. Towards self-configuration in autonomic electronic institutions. To appear In *Coordination, Organization, Institutions and Norms in agent systems*, Lecture Notes in Computer Science.Springer Verlag, 2007.
- [Bou et al., 2008] Eva Bou , Maite López-Sánchez and Juan Antonio Rodríguez-Aguilar, Using Case-Based Reasoning in Autonomic Electronic Institutions. *Coordination, Organizations, Institutions, and Norms in Agent Systems III*, Lecture Notes in Computer Science. Editor Springer Berlin / Heidelberg. ISSN 0302-9743 (Print) 1611-3349 (Online). Volume 4870/2008. DOI 10.1007/978-3-540-79003-7. ISBN 978-3-540-79002-0. Páginas 125-138
- [Bou et al., 2009] Eva Bou, Maite López-Sánchez, Juan A. Rodríguez-Aguilar, Jaime Simao Sichman. Adapting Autonomic Electronic Institutions in Heterogeneous Agent Societies. In *Organised*
- [Bratman, 1987] Bratman M.E. (1987). *Intentions, Plans and Practical Reason*. Harvard University Press, Cambridge, M.A.
- [Bratman et al., 1988] Bratman M.E., Israel D. and Pollack M.E. (1988). Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4. pages 349-355.
- [Brenner et al., 1998] Brenner W., Zarnekow R., Wittig H. "Intelligent Software Agents : Foundations and Applications" Springer 98.
- [Bresciani et al., 2004] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia and J. Mylopoulos. 2004. Tropos: An Agent-Oriented Software Development Methodology, *Autonomous Agents and Multi-Agent Systems* 8 (3): 203-236. doi:http://dx.doi.org/10.1023/B:AGNT.0000018806.20944.ef.

## Referencias

---

- [Brogi et al., 2003] Brogi A, Corfini S, Popescu R (2003) Composition-oriented service discovery. Proceedings of 5th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)
- [Brooks, 1986] Brooks, R.: A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation 2(1), 1986, pp. 14 – 23.
- [Brooks, 1991] Brooks R. A. (1991). Intelligence without Representation. Artificial Intelligence, 47, 139-159.
- [Burt, 1987] R. Burt. Social contagion and innovation: Cohesion versus structural equivalence. American J. of Sociology, 92:1287–1335, 1987.
- [Busi et al., 2001] Busi, N.; Ciancarini, P.; Gorrieri, R.; y Zavattaro, G. (2001): Coordinations Models: A Guide Tour. Coordination of Internet Agents: Models, Technologies, and Applications (Omicini, Zambonelli, Klush y Tolksdorf, editors). Springer-Verlag, 6-24.
- [Caicedo y Thalmann, 2000] Caicedo, A, Thalmann, D: "Virtual Humanoids: Let Them Be Autonomous without Losing Control", 3IA International Conference. 4ta ed. (2000)
- [Campos et al., 2009] J. Campos, M. Lopez, M. Esteva. Norm adaptation in 2-LAMA: a Two Level Adaptive Multiagent system Architecture. Submitted to the Special issue on "Normative Multiagent Systems" in the Logic Journal of the IGPL, 2009.
- [Capera et al., 2003] Capera, D., Georgé, J.-P., Gleizes, M.-P., & Glize, P. (2003). Emergence of organisations, emergence of functions. Symposium on Adaptive Agents and Multi-Agent Systems, (págs. 103-108).
- [Carbonell, 1986] Carbonell J.G. Derivational analogy: A theory of constructive problem solving and expertise acquisition. En Michalski, RS, Carbonell, JG & Mitchell, TM (eds.) Machine Learning: An Artificial Intelligence Approach (Volume II), Los Altos, CA: Morgan Kaufmann, pp. 371 – 392.
- [Carrascosa et al., 2003] C. Carrascosa, M. Rebollo, J. Soler, V. Julian and V. Botti SIMBA Architecture for Social Real-Time Domains EUMAS 2003: The First European Workshop on Multi-Agent Systems pp. 0-0. (2003)
- [Carrascosa et al., 2008] Carrascosa C., Bajo J., Julián V., Corchado J.M. y Botti V.: Hybrid multi-agent architecture as a real-time problem-solving model. Expert Systems with Applications, vol. 34 (1) pp. 2-17. Pergamon-Elsevier Science LTD. doi:10.1016/j.eswa.2006.08.031. (2008)
- [Carrascosa et al., 2009] C. Carrascosa, A. Giret, V. Julian, M. Rebollo, E. Argente, V. Botti .2009. Service Oriented MAS: An open architecture (Short Paper), Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009), Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary: 1291–1292
- [Casillas, 2004] Casillas, T. Guía para la creación de empresas. 2004, Fundación EOI: Madrid
- [Castelfranchi y Conte, 1995] Castelfranchi C. Conte R. The treatment of norms in modelling rational agents: cognitive issues. In Proceedings of ModelAge 95, General Meeting of Esprit Working Group 8319, INRIA France, 1995.

- 
- [Castelfranchi, 2000] Castelfranchi, C.: Engineering Social Order. In: Omicini, A., Tolksdorf, R., Zambonelli, F. (Eds.): Engineering Societies in the Agents World, LNAI 1972, Springer-Verlag, 2000, pp. 1 – 19.
- [Castillo, 1998] Castillo L. Desarrollo y aplicación de técnicas de planificación no lineal para la programación del control de plantas industriales (Development and application of nonlinear planning techniques for control programming in manufacturing systems). PhD thesis, 1998
- [Cavedon y Rao, 1996] Cavedon, L., & Rao, A. S. (1996). Bringing about rationality: Incorporating plans into a BDI agent architecture. In N. Y. Foo, & R. Goebel (Ed.), In Proceedings of the 4th Pacific Rim international Conference on Artificial intelligence: Topics in Artificial intelligence. Lecture Notes In Computer Science (LNCS). 1114. Springer-Verlag, London, UK.
- [Cetkovic y Parmee , 2002] Cvetkovic D. y Parmee I. Agent-based support within an interactive evolutionary design system. Artificial Intelligence for Engineering Design, Analysis and Manufacturing vol. 16 (5), pp. 331—342. (2002)
- [Cimatti y Roveri, 2000] Cimatti A. y Roveri M. Conformant Planning via Symbolic Model Checking. Journal of Artificial Intelligence Research, vol. 13, pp: 305-338. 2000
- [Cohen y Levesque, 1990] Cohen, P., Levesque H.: Intention is Choice with Commitment. Artificial Intelligence 42(2-3), March 1990, pp. 213-261.
- [Conde y Thalmann, 2006a] Toni Conde, Daniel Thalmann: Learnable behavioural model for autonomous virtual agents: low-level learning. AAMAS 2006: 89-96
- [Conde y Thalmann, 2006b] Toni Conde, Daniel Thalmann: An integrated perception for autonomous virtual agents: active and predictive perception. Journal of Visualization and Computer Animation 17(3-4): 457-468 (2006)
- [Conte y Paolucci, 2001] R. Conte and M. Paolucci. Intelligent social learning. Artificial Society and Social Simulation, 4(1):1-23, 2001.
- [Corchado, 2004] Corchado J.M. (2004). Agentes Software y sistemas multi-agente. Pérez de la Cruz J.L. In press. Pearson Educación.
- [Corchado y Laza, 2003] Corchado, J. M., & Laza, R. (2003). Constructing Deliberative Agents with Case-based Reasoning Technology. International Journal of Intelligent Systems , 18 (12), 1227-1241.
- [Corchado et al., 2005] Corchado, J. M., Pavón, J., Corchado, E., & Castillo, L. F. (2005). Development of CBR-BDI agents: A tourist guide application. In Proceedings of the 7th European Conference on Case-based Reasoning 2004, Lecture Notes in Artificial Intelligence (LNAI). 3155, pp. 547-559. Springer-Verlag.
- [Corchado et al., 2006] Corchado, J. M., Bajo, J., De Paz, Y. & Tapia, D. I. (2006). Intelligent Environment for Monitoring Alzheimer Patients, Agent Technology for Health Care. Decision Support Systems. Elsevier Science. Amsterdam, Netherlands.
- [Corchado et al., 2007] Corchado J.M., Aiken J. and Bajo J.: A CBP Agent for Monitoring the CO2 Exchange Rate. Book Chapter in Case-Based Reasoning on Signals and Images. Petra Perner (Ed.). Studies in Computational Intelligence vol. 73, ISBN: 978-3540731788, Springer Verlag. In Press. (2007)

## Referencias

---

- [Corchado et al. 2008] Corchado J. M. , Gonzalez-Bedia M. , De Paz Y. , Bajo J. y De Paz J.F.: Replanning mechanism for deliberative agents in dynamic changing environments. Computational Intelligence ISSN: 0824-7935. Mayo 2008. Volumen 24, núm. 2, Pág. 77-107 DOI: 10.1111/j.1467-8640.2008.00323.x
- [Cossentino, 2005] M. Cossentino. 2005. From requirements to code with the passi methodology, Agent Oriented Methodologies IV: 79–106.
- [Cossentino et al., 2001] Cossentino, M., Potts, C.: Passi: A process for specifying and implementing multi-agent systems using uml. Tech. rep., Technical report, University of Palermo (2001)
- [Cox et al., 2006] Cox M. T., Muñoz-Avila H., y Bergmann R. (2006). Case-based planning. Knowledge Engineering Review. 20(3): 283-287.
- [Criado et al., 2009]. N. Criado, E. Argente, V. Julian and V. Botti. 2009. Designing Virtual Organizations. 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS2009). Volume 55:440-449.
- [Currie and Tate, 1991] Ken Currie and Austin Tate. O-Plan: the open planning architecture. Artificial Intelligence, 52(1):49–86, 1991.
- [Dautenhahn, 2000] Dautenhahn, K.: Reverse Engineering of Societies - A Biological Perspective. Proc. of AISB Symposium "Starting from Society - application of social analogies to computational systems". Convention of the Society for the Study of Artificial Intelligence and the Simulation of Behavior (AISB-00), University of Birmingham, England, 2000.
- [Davidsson, 2001] Davidsson, P.: Categories of Artificial Societies. In: A. Omicini, P. Petta, R. Tolksdorf (Eds.): Engineering Societies in the Agents World II, Springer Verlag LNAI 2203, 2001.
- [De Paz et al., 2009] J.F. De Paz, S. Rodríguez, J. Bajo and J.M. Corchado. Mathematical model for dynamic case-based planning. International Journal of Computer Mathematics. ISSN: 0020-7160. Noviembre 2009. Vol 86, núm 10–11, Pág. 1719–1730. DOI:10.1080/00207160802562556
- [Dellarocas, 2000] Dellarocas, C.: Contractual Agent Societies: Negotiated Shared Context and Social Control in Open Multi-agent Systems. In: Proc. of Workshop on Norms and Institutions in Multi-Agent Systems, Autonomous Agents-2000, Barcelona, 2000.
- [DeLoach, 2009] S. DeLoach. 2009. Multi-Agent Systems: Semantics and Dynamics of Organizational Models, IGI Global, Ch. Organizational Model for Adaptive Complex Systems: 1–26.
- [Di Marzo et al., 2004] Di Marzo Serugendo, G., Gleizes, M.-P., & Karageorgos, A. (2004). AgentLink First Technical Forum Group Self-Organisation in Multi-Agent Systems. AgentLink Newsletter, Issue 16, 23-24.
- [Dignum, 2001] Dignum, F.: Agents, Markets, Institutions and Protocols. In: Dignum, F., Sierra, C. (Eds.): Agent Mediated Electronic Commerce, LNAI 1991, Springer, 2001, pp. 98 – 114.
- [Dignum, 2004] V. Dignum. A model for organizational interaction: based on agents, founded in logic, PhD. Thesis, 2004.



- [Dignum, 2006] Dignum V, Dignum F (2006) A landscape of agent systems for the real world. Technical report 44-cs-2006- 061, Institute of Information and Computing Sciences, Utrecht University
- [Dignum y Dignum, 2007] V. Dignum and F. Dignum. 2007. A logic for agent organization. In Proc. FAMAS@Agents'007.
- [Dignum et al., 2001] Dignum, V., Weigand, H., Xu L.: Agent Societies: Towards Framework-Based Design. In: Wooldridge, M., Ciancarini P., Weiss, G. (Eds.): Agent-Oriented Software Engineering II. LNAI 2222, Springer, 2001, pp. 33 - 49.
- [Dignum et al., 2005] V. Dignum, J. Vazquez-Salceda, and F. Dignum. 2005. Omni: Introducing social structure, norms and ontologies into agent organizations. LNAI 3346.
- [Dominguez et al., 1995] Domínguez Machuca, J.A. et al.: "Dirección de Operaciones" MacGraw Hill 1995 503p.
- [Doran et al., 1997] J. E. Doran, N. Franklin, N. R. Jennings, and T. Norman. On cooperation in multi-agent systems. The Knowledge Engineering Review, 12(3):309-314, 1997.
- [D'Inverno y Luck, 2004] D'Inverno M. y Luck M. Understanding Agent Systems. Springer Verlag. isbn 3540407006. 2004.
- [Dumas et al., 1995] Yvan Dumas, Jacques Desrosiers, Eric Gelinás and Marius M. Solomon. An Optimal Algorithm for the Traveling Salesman Problem with Time Windows. Operations Research, Vol. 43, No. 2 (Mar. - Apr., 1995), pp. 367-371 Published by: INFORMS
- [Dunin-Keplicz y Verbrugge, 2003] Dunin-Keplicz, B., Verbrugge, R: Calibrating Collective Commitments. In: Marik, V., Müller, J., Pechoucek, M. (Eds.): Multi-Agent Systems and Applications, LNAI 2691, Springer, 2003, pp. 73 - 83.
- [Duong y Grefenstette, 2005] D. V. Duong, J. Grefenstette, "The emulation of social institutions as a method of coevolution" in GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation, ACM Press, pp. 555-556, 2005.
- [Edmonds, 1999] Edmonds, B.: Capturing Social Embeddedness: a Constructivist Approach. Adaptive Behavior 7, 1999, pp. 323-348.
- [Escriva et al., 2006] M. Escrivá, J. Palanca, G. Aranda, A. GarcíaFornes, et al. A Jabberbased multiagent system platform. In Proc. of AAMAS06, 12821284, 2006
- [Esteva, 2003] Marc Esteva. Electronic Institutions: from specification to development Ph. D. Thesis, Technical University of Catalonia, 2003.
- [Esteva et al., 2001] Esteva, M., Rodríguez, J., Sierra, C., Garcia, P., Arcos J.: On the formal specifications of electronic institutions, In: Dignum F., Sierra C. (Eds.) Agent-mediated Electronic commerce (The European AgentLink Perspective), LNAI 1991, Springer, 2001, pp. 126-147.
- [Esteva et al., 2002] Esteva, M., de la Cruz, D., Sierra, C.: ISLANDER: an Electronic Institutions Editor. In: C. Casterfranchi, W. L. Johnson (Eds.): Proc. of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002), Bologna, Italy, 2002, pp. 1045-1052.

## Referencias

---

- [Ferber, 1999] Jacques Ferber. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley Longman Publishing Co., Boston, MA, USA, 1999.
- [Ferber, Gutknecht, 1998] Ferber, J., Gutknecht, O.: A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems. In: Proc. of the 3rd. International Conference on Multi-Agent Systems (ICMAS'98), IEEE Computer Society, 1998.
- [Ferber et al., 2004] Ferber, O. Gutknecht, F. Michel. 2004. From Agents to Organizations: an Organizational View of Multi-Agent Systems, in: P. Giorgini, J. Muller, J. Odell (Eds.), *Agent-Oriented Software Engineering VI*, Vol. LNCS 2935 of Lecture Notes in Computer Science, Springer-Verlag: 214–230.
- [Fergusson, 1995] Fergusson, I.: Integrated Control and Coordinated Behavior: A case for agent models. M. Wooldridge, N. Jennings (Eds.): *Intelligent Agent: Theories, Architectures and Languages*, LNAI 890, Springer, 1995, pp. 203 – 218.
- [Ferrel y Hirt, 2004] Ferrer O. C., H.G., ed. *Introducción a los negocios en un mundo cambiante*. 2004, McGraw Hill. 177.
- [Fikes y Nilsson, 1971] Fikes,R.E. y Nilson,N.J. STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189-208, 1971
- [Fischer, 1994] Fischer, M.: A Survey of METATEM, the Language and its Applications. In:Gabbay, D., Ohlbach, H. (Eds.): *Proceedings of the 1st. International Temporal Logic Conference*, LNAI 827, Springer, 1994, pp. 480 – 505.
- [Fonseca et al., 2001] S.P. Fonseca, Martin L. Griss, Reed Letsinger, *An Agent Mediated E-Commerce Environment for the Mobile Shopper*, Hewlett-Packard Laboratories, Technical Report, HPL-2001-157, July 2001.
- [Foster et al., 2001] I. Foster, C. Kesselman, and S. Tuecke. 2001. The anatomy of the grid: Enabling scalable virtual organizations, *Int. J. High Perform. Comput. Appl.*, vol. 15, no.~3: 200--222
- [Fox y Gruniger, 1998] Fox, M., Gruninger, M.: *Enterprise Modeling*. AI Magazine, AAAI Press, Fall, 1998, pp. 109-121.
- [Franklin y Graesser, 1997] Franklin S. y Graesser A. "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents," *Proceedings of the Agent Theories, Architectures, and Languages Workshop*, Berlin: Springer Verlag, 193-206. (1997)
- [Garcia, 2009] A. Garcia Camino. *Normative Regulation of Open Multi-agent Systems*. PhD thesis, Universidad Autonoma deBarcelona, 2009.
- [Garca-Magario et al. 2009] Garca-Magario, I., Gómez-Sanz, J., Fuentes, R.: Ingenias development assisted with model transformation by-example: A practical case. In: 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009), pp. 40 – 49 (2009)
- [Gasser et al., 1987] L. Gasser, C. Braganza, and N. Herman. *Distributed Artificial Intelligence*, chapter MACE: A flexible test-bed for distributed AI research, pages 119–152. Pitman Publishers, 1987.
- [Gasser y Ishida, 1991] Gasser, L., Ishida, T.: A dynamic organizational architecture for adaptive problem solving. In: Proc. of AAAI-91, 185–190, 1991

- 
- [Gateu et al., 2005] Gateau, B., Boissier, O., Khadraoui, D., Dubois, E.: Moiseinst: An organizational model for specifying rights and duties of autonomous agents. In: Third European Workshop on Multi-Agent Systems (EUMAS 2005), pp. 484–485 (2005)
- [Georgeff y Lansky, 1987] Georgeff M. y Lansky A. L. (1987). Reactive reasoning and planning. Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87). Seattle, WA, 1987.
- [Georgeff et al., 1999] Georgeff M., Pollack M. y Tambe M. (1999). The Belief-Desire-Intention Model of Agency. Proceedings of the Fifth International Workshop on Agent Theories, Architectures and Languages (ATAL-98), Lecture Notes in Artificial Intelligence. Springer-Verlag, Heidelberg, 1999.
- [Genesereth y Nilsson, 1987] Genesereth, M., Nilsson, N.: Logical Foundations of Artificial Intelligence, Morgan Kaufman, 1987.
- [Ghallab y Laruelle, 1994] Ghallab M. y Laruelle H. Representation and control in IxTeT, a temporal planner. En Pro. 2nd Int. Conference on AI Planning Systems, pág 61-67. Hammond, 1994
- [Giampapa y Sycara, 2002] J. A. Giampapa and K. Sycara, "Team-Oriented Agent Coordination in the RETSINA Multi-Agent System," Tech. Report CMU-RI-TR-02-34, Robotics Institute, Carnegie Mellon University, December 2002. Presented at AAMAS 2002 Workshop on Teamwork and Coalition Formation.
- [GTI-IA, 2009] GTI-IA. An Abstract Architecture for Virtual Organizations: The THOMAS project. <http://www.fipa.org/docs/THOMASarchitecture.pdf>
- [Giret et al., 2005] Giret A., Botti V. and Valero, S.. MAS Methodology for HMS. Proceedings of the Second International Conference on Applications of Holonic and Multi-Agent Systems, HoloMAS 2005 Vol. LNAI 3593 pp. 39-49. (2005)
- [Giret et al., 2009] Giret, V. Julian, M. Rebollo, E. Argente, C. Carrascosa and V. Botti. 2009. An Open Architecture for Service-Oriented Virtual Organizations. Seventh international Workshop on Programming Multi-Agent Systems.PROMAS 2009: 23-33.
- [Glez-Bedia, 2004] Glez-Bedia M. (2004). Fundamentos cognitivos para el diseño de arquitecturas de agentes planificadores en contextos dinámicos de acción. Tesis Doctoral. Universidad de Salamanca.
- [Golden et al., 1987] B Golden, L Levy, R Vohra. The orienteering problem. Naval Research Logistics. 1987
- [Golding y Rosenbloom, 1988] Golding A.R. y Rosenbloom P.S. Combining Analytical and Similarity-Based {CBR}, pp. 259-263.1988
- [Grosz y Kraus, 1996] Grosz, B., Kraus, S.: Collaborative Plans for Complex Group Actions. Artificial Intelligence 86, 1996, pp. 269-358.
- [Gutknecht y Ferber, 1997] Gutknecht, O. and Ferber, J. 2000. MadKit: a generic multi-agent platform. In Proceedings of the Fourth international Conference on Autonomous Agents (Barcelona, Spain, June 03 - 07, 2000). AGENTS '00. ACM, New York, NY, 78-79. DOI= <http://doi.acm.org/10.1145/336595.337048>

## Referencias

---

- [Hammond, 1989] Hammond K.J. Case-Based Planning: Viewing Planning as a Memory Task. Academic Press Inc., 1989.
- [Hales, 2002] Hales, D. Group Reputation Supports Beneficent Norms. *The Journal of Artificial Societies and Social Simulation (JASSS)* vol. 5, no. 4. 2002.
- [Hahn et al., 2008] Hahn, C., Madrigal-Mora, C., Fischer, K.: A platform-independent metamodel for multiagent systems. *Autonomous Agents and Multi-Agent Systems* 18(2), 239–266 (2008).
- [Hayes-Roth, 1988] Hayes-Roth, B. (1988). Making Intelligence systems adaptive. Report STAN-CS-88-1226, Stanford University, Stanford, CA, October, 1988; Hayes-Roth, B. (1990). Architectural foundations for real-time performance in intelligent agents. *The Journal of Real-Time Systems*, 2, pp.99-125,1990.
- [Held y Karp, 2005] Michael Held and Richard M. Karp. The traveling-salesman problem and minimum spanning trees: Part II. *Mathematical Programming*. Springer Berlin / Heidelberg. ISSN 0025-5610 (Print) 1436-4646 (Online) Volume 1, Number 1 / diciembre de 1971. DOI 10.1007/BF01584070. Páginas 6-25. Received: 19 October 1970. Mayo de 2005
- [Hernández, 2001] Hernández Matías, J.C.; "Metodología para el análisis y planificación de acciones de mejora continua en sistemas de fabricación". PhD Tesis, Universidad Politécnica de Madrid, 2001.
- [Hernández et al. 2006] L. Hernandez, V. Botti, A. Garcia-Fornes. A deliberative scheduling technique for a real-time agent architecture. *Engineering Applications of Artificial Intelligence* Volume: 19 Issue: 5 Pages: 521-534 Published: AUG 2006
- [Hinze y Voisard, 2003] A. Hinze and A. Voisard. Location- and time-based information delivery in tourism. In *Advances in Spatial and Temporal Databases (SSTD 2003)*, volume 2750 of LNCS, July 2003.
- [Hinze et al., 2005] A. Hinze, R. Malik, and P. Malik. Towards a TIP 3.0 Service-Oriented Architecture: Interaction Design. Technical Report Number 08/2005, Department of Computer Science, University of Waikato, August 2005
- [Hodge y Gales, 2003] Hodge, B.A., W.; y Gales, L., *Teoría de la Organización: un enfoque estrategico*. 2003:Pearson Educación
- [Horling y Lesser, 2004] B. Horling and V. Lesser. 2005. Using ODML to Model Multi-Agent Organizations. In *Proc. of the IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology*
- [Howden et al., 2001] Howden, N., et al. (2001). JACK intelligent agents—summary of an agent infrastructure. In *Proceedings of IEEE international conference on autonomous agents*, Montreal.
- [Hubner, 2004] Hubner, J.F., Sichman, J.S., Boissier, O.: Using the Moise+ for a cooperative framework of mas reorganisation. In: *LNAI Proc. of the 17th Brazilian Symposium on Artificial Intelligence (SBIA'04)*. Volume 3171, 506–515, Springer , 2004.
- [Hubner et al., 2006] J. Hubner, J. Sichman, O. Boissier. 2006. S-Moise+: A middleware for developing organised multi-agent systems. In: *Proc. Int. Workshop on*

- Organizations in Multi-Agent Systems, from Organizations to Organization Oriented Programming in MAS, LNCS, vol 3913: 64–78
- [Huhns y Stephens, 1999] Huhns, M., Stephens, L.: Multiagent Systems and Societies of Agents. In: Weiss, G. (Ed.), Multi-agent Systems: a Modern Approach to Distributed Artificial Intelligence, MIT Press, 1999.
- [Hung-Chen y Arbee, 2005] Hung-Chen Chen and Arbee L. P. Chen. A Music Recommendation System Based on Music and User Grouping. Journal of Intelligent Information Systems. Springer Netherlands ISSN 0925-9902 (Print) 1573-7675 (Online). Volume 24, Numbers 2-3 Páginas 113-132. Marzo de 2005. 10.1007/s10844-005-0319-3
- [Hyacinth et al., 1999] Hyacinth S. Nwana, Divine T. Ndumu, Lyndon C. Lee, Jaron C. Collis: ZEUS: A Toolkit and Approach for Building Distributed Multi-Agent Systems. Agents 1999: 360-361
- [Iglesias, 2010] José Antonio Iglesias Martínez. Modelado automático del comportamiento de agentes inteligentes. Phd Tesis, . Departamento de Informática. Universidad Carlos III de Madrid. 2010.
- [Jennings, 1993] Jennings, N. R.(1993). Specification and implementation of a belief-desire-jointintention architecture for collaborative problem solving. Intl. Journal of Intelligent and Cooperative Information Systems 2(3) 289–318
- [Jennings y Wooldridge, 1998] Jennings, N., Wooldridge, M. (eds.), Agent Technology: Foundations, Applications and Markets, Springer, 1998.
- [Jennings y Wooldridge, 2002] N. R. Jennings and M. Wooldridge. 2002. Agent-oriented software engineering. Handbook of Agent Technology.
- [Jennings et al., 1998] N. R. Jennings, S. Parsons, P. Noriega, and C. Sierra. On argumentation-based negotiation. In International Workshop on Multi-Agent Systems, 1998.
- [Jensen y Veloso, 2000] Jensen R.M. y Veloso M.M. OBDD-Based Universal Planning for Multiple Synchronized Agents in Non-Deterministic Domains. Artificial Intelligence Planning Systems, pp:167-176. 2000
- [Jiao y Mei, 2004] W.P. Jiao, Mei H. Automated adaptations to dynamic software architectures by using autonomous agents. Engineering Applications of Artificial Intelligence Volume: 17 Issue: 7 Pages: 749-770 Published: OCT 2004
- [Jonker et al., 2000] Jonker, C., Klusch, M., Treur, J.: Design of Collaborative Information Agents. In: Klusch M., Kerschberg, L. (eds.): Cooperative Information Agents IV. LNAI 1860, Springer-Verlag, 2000, pp. 262 – 283.
- [Juan y Sterling, 2002] Juan, T.P., A.; Sterling, L. Roadmap: Extending the Gaia methodology for complex open systems. in 1st Int. Joint Conference on Autonomous Agents and multiagent Systems (AAMAS 2002). 2002.
- [Kalenka y Jennings, 1999] S. Kalenka and N. R. Jennings. Cognition, Agency and Rationality, chapter Socially Responsible Decision Making by Autonomous Agents, pages 135–149. Kluwer, 1999.
- [Foster y Kesselman, 2004] I. Foster and C. Kesselman, editors. The Grid 2: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 2004.

## Referencias

---

- [Kautz y Selman, 1996] Kautz H. y Selman B. Pushing the envelope: Planning, propositional logic and stochastic search. En Proc. 13th Nat. Conference on AI, pág 1194-1201. 1996.
- [Kautz et al., 2006] Kautz H., Selman B. y Hoffmann J. SatPlan: Planning as Satisfiability Abstracts of the 5th International Planning Competition, 2006
- [Kittock 1993] J. E. Kittock. Emergent conventions and the structure of multi-agent systems. In Lectures in Complex systems: the proceedings of the 1993 Complex systems summer school, Santa Fe Institute Studies in the Sciences of Complexity Lecture Volume VI, Santa Fe Institute, pages 507-521. Addison-Wesley, 1993.
- [Klusch y Sycara, 2001] Klusch, M.; y Sycara, K. (2001): Brokering and matchmaking for coordination of agent societies: a survey. En Coordination of Internet Agents: Models, Technologies, and Applications (Omicini et al.) Springer, 197-224
- [Knoblock, 1993] Knoblock, Craig; "Generating Abstraction Hierarchies- An automated approach to reducing search in planning". Kluwer, Dordrecht Holanda, 1993.
- [Kolodner, 1993] Kolodner J. (1993). Case-Based Reasoning. San Mateo, Morgan Kaufman.
- [Kowalczyk et al., 2002] R. Kowalczyk, M. Ulieru and R. Unland, Integrating Mobile and Intelligent Agents in Advanced e-Commerce: A Survey, in Proc. of Agent Technologies, Infrastructures, Tools, and Applications for E-Services, NODe 2002, Lecture Notes in Computer Science, Vol. 2592 (Springer, Berlin, 2002) 295-313.
- [Labidi y Lejouad, 1993] Labidi, S., Lejouad, W. (1993) Del'intelligence artificielle distribu'eeaux syst'emes multiagents
- [Labordere, 1969] A. L. Henry-Labordere (1969) The record balancing problem: a dynamic programming solution of a generalized travelling salesman problem. RIRO B 2, 43-49
- [Lakkaraju y Gasser, 2008] Kiran Lakkaraju and Les Gasser. "Norm Emergence in Complex Ambiguous Situations." In Proceedings of the AAAI Workshop on Coordination, Organizations, Institutions and Norms AAAI, Chicago, July, 2008.
- [Leake et al., 2000] Leake D.B., Bauer T., Maguitman A. y Wilson D.C. Capture, Storage and Reuse of Lessons about Information Resources: Supporting Task-Based Information Search. Proceedings of the AAAI-00 Workshop on Intelligent Lessons Learned Systems. Austin, Texas, pp. 33-37. 2000
- [Lee, 1997] Lee J.M. Riemannian Manifolds. An introduction to Curvature. Springer-Verlag, New York, Inc. 1997
- [Lésperance et al., 1996] Lésperance, Y., Levesque, H., Lin, F., Marcu, D., Reiter, R., Scherl, R.: Foundations of a Logical Approach to Agent Programming. In: Wooldridge, M., Müller, J., Tambe, M. (Eds.): Intelligent Agents II, LNAI 1037, Springer, 1996, pp. 331 - 346.
- [López de Mántaras, 2001] Ramon López de Mántaras: Case-Based Reasoning. Machine Learning and Its Applications 2001: 127-145
- [López de Mántaras y Plaza, 1997] López de Mántaras R., Plaza E. Case-Based Reasoning: An overview. AI Communications Journal 10(1) (1997) 21-29.

- 
- [López de Mántaras et al., 2005] López De Mántaras R., McSherry D., Bridge D., Leake D., Smyth B., Craw S., Faltings B., Maher M.L., Cox M.T., Forbus K., Keane M., Aamodt A. y Watson I. Retrieval, reuse, revision and retention in case-based reasoning. *The Knowledge Engineering Review*, vol. 20 (3), pp.215-240. 2005
- [López et al., 2006] F. Lopez, M. Luck, and M. d'Inverno. 2006. A normative framework for agent-based systems. *Computational and Mathematical Organization Theory*, 12:227–250.
- [Lozano y Calderón, 2006] Lozano, M., Calderón, C.: Entornos Virtuales 3D clásicos e inteligentes: hacia un marco de simulación para aplicaciones gráficas interactivas (2006)
- [Luck et al., 2005] M. Luck and et al. Agent technology roadmap. M. Luck, P. McBurney, O. Sherhory, S. Willmott, 2005.
- [Luck et al., 2008] M. Luck, P. McBurney. 2008. Computing as interaction: Agent and agreement technologies, in: *IEEE SMC Conference on Distributed Human-Machine Systems*: 1–6.
- [Maes, 1990] Maes, P. (Ed.): *Designing Autonomous Agents*. MIT Press, 1990.
- [Malone, 1988] T. W. Malone. What is coordination theory? In *National Science Foundation Coordination Theory Workshop*. MIP, EE.UU., 1988.
- [Malone y Crowston, 1994] Malone, T., Crowston, K.: *The Interdisciplinary Study of Coordination*. *ACM Computing Surveys* 26(1), March, 1994.
- [Mas, 2005] Mas, A. *Agentes software y sistemas multiagente : conceptos, arquitecturas y aplicaciones*. ISBN 84-205-4367-5 , pp. 29-64. 2005
- [Mataric, 1997] M.J. Mataric, "Learning Social Behavior" in *Robotics and Autonomous Systems*, vol. 20, pp. 191-204, 1997
- [Mauro, 2003] P.P. Mauro "The Emergence of Social Learning in Artificial Societies" in *Applications of Evolutionary Computing: EvoWorkshops 2003: EvoBIO, EvoCOP, EvoIASP, EvoMUSART, EvoROB, and EvoSTIM*, Essex, UK, pp. 467-478, 2003.
- [McAllester y Rosenblitt, 1991] D. McAllester and D. Rosenblitt. Systematic nonlinear planning. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, volume 2, pages 634--639, Anaheim, California, USA, 1991. AAAI Press/MIT Press
- [McCarthy, 1978] McCarthy J. Ascribing mental qualities to machines. Technical report, Stanford University AI Lab., Stanford, CA 94305. 1978
- [Melville et al., 2002] Prem Melville and Raymond J. Mooney and Ramadass Nagarajan. : Content-Boosted Collaborative Filtering for Improved Recommendations. *Proceedings of the Eighteenth National Conference on Artificial Intelligence(AAAI-2002)*,pp. 187-192, Edmonton, Canada, July 2002
- [Mooney y Roy, 2000] Raymond J. Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, pages 195{204. ACM Press, 2000..
- [Noriega, 1999] P. Noriega, *Agent-Mediated Auctions: The Fishmarket Metaphor*. Monografías of IIIA No. 8. Barcelona (1999).



## Referencias

---

- [Noriega y Sierra, 2002] P. Noriega, C. Sierra. 2002. Electronic Institutions: Future Trends and Challenges. CIA: 14-17
- [North, 1990] D. North. Institutions, Institutional Change and Economics Performance. Cambridge U. P., 1990.
- [Nwana, 1995] Nwana H. S. Software Agents: An Overview. Knowledge Engineering Review, vol. 11 (2), pág 205-244. (1995)
- [Molesini et al., 2006] A. Molesini, A. Omicini, E. Denti, A. Ricci. SODA. 2006. A roadmap to artefacts, Engineering Societies in the Agents World VI LNAI 3963: 49-62.
- [Müller et al., 1995] Müller, J., Pischel, M., Thiel, M.: Modelling Reactive Behavior in Vertically Layered Agent Architectures. In: Wooldridge, M., Jennings, N. (Eds.): Intelligent Agent: Theories, Architectures and Languages, LNAI 890, Springer, 1995, pp. 261 - 276.
- [Muller, 1996] H. J. Muller. Negotiation principles. In John Wiley & Sons, editor, Foundations of Distributed Artificial Intelligence, pages 211{230, Norway,1996.
- [Newell y Simon, 1963] Newell A. y Simon H. GPS, a Program that Simulates Human Thought. McGraw Hill, NY, 1963.
- [Omicini et al., 2004] A. Omicini, A. Ricci, M. Viroli. Coordination artifacts: Environment-based coordination for intelligent agents. In Proceedings of 3rd international Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), 286—293, 2004.
- [Ossowski, 1998] Ossowski, S.: Co-ordination in Artificial Agent Societies, LNAI 1535, Springer, 1998.
- [Ossowski, 2001] Ossowski, S. (2001): Agent Coordination by Constraint Optimisation. Electronic Notes in Theoretical Computer Science 48, Elsevier.
- [Padgham, 2002] Padgham, L.W., M. Prometheus: A methodology for developing intelligent agents. in Workshop on Agent Oriented Software Engineering (AOSE-2002). 2002.
- [Parunak y Odell, 2002] Parunak, H. V. D., Odell, J.: Representing Social Structures in UML. In: Wooldridge, M., Weiss, G., Ciancarini P. (Eds.): Agent-Oriented Software Engineering II, LNCS 2222, Springer, 2002.
- [Pattison et al., 1987]. H. E. Pattison, D. D. Corkill, and V. R. Lesser. Distributed Artificial Intelligence, chapter Instantiating Descriptions of Organizational Structures, pages 59-96. Pitman Publishers, 1987.
- [Pavón y Gomez-Sanz, 2003] J. Pavon and J. J. Gomez-Sanz. Agent oriented software engineering with INGENIAS. In Proceedings of CEECMAS, volume 2691, pages 394-403, 2003.
- [Pavón et al, 2005] J. Pavon, J. Gomez-Sanz, R. Fuentes. 2005. The INGENIAS Methodology and Tools, Idea Group Publishing, article IX: 236-276.
- [Peiro, 1990] Peiró, J. M. (1990): «Las Nuevas Tecnologías» en Organización. Nuevas perspectivas psicosociales, Barcelona, PPU S.A.



- [Penberthy y Weld, 1992] Penberthy, J. S. and Weld, D., UCPOP: A Sound, Complete, Partial-Order Planner for ADL. Third International Conference on Knowledge Representation and Reasoning (KR-92), Cambridge, MA, October 1992.
- [Perini y Susi, 2006] Perini, A., Susi, A.: Automating model transformations in agent-oriented modelling. In: Agent-Oriented Software Engineering VI, pp. 167 - 178 (2006)
- [Popper, 1982] Popper, K.: The Open Universe: An Argument for Indeterminism, Hutchinson, London, 1982.
- [Prantner , 2004] K. Prantner. Ontour: The ontology. Technical report, DERI Innsbruck, 2004. <http://e-tourism.deri.at>
- [Rao y Georgeff, 1991] Rao, A. S., & Georgeff, M. P. (1991). Modeling rational agents within a BDI-Architecture. In J. Allen, R. Fikes, & E. Sandewall (Ed.), Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91) (pp. 473-484). San Mateo, CA, USA: Morgan Kaufmann publishers, Inc
- [Rao y Georgeff, 1992] Rao, A., Georgeff, M.: An Abstract Architecture for Rational Agent. In: C. Rich, W. Swartout, B. Nebel (Eds.): Proc. of KR'92, Morgan Kaufmann, 1992, pp. 439 - 449.
- [Rao y Georgeff, 1995] Rao, A. S., & Georgeff, M. P. (1995). BDI agents: from theory to practice. In Proceedings of the First International Conference on Multi-Agents Systems (ICMAS'95). San Francisco, CA, USA.
- [Rao y Georgeff, 1998] Rao A. S. and Georgeff M. P. (1998). Decision procedures of BDI. Logics. Journal of logic and computation 8(3). 1998.
- [Razavi et al., 2005] Razavi, R., Perrot, J., & Guelfi, N. (2005). Adaptive modeling: an approach and a method for implementing adaptive agents. Lecture Notes in Artificial Intelligence , 3446, 136-148.
- [Refanidis y Valvas, 2001] Refanidis I. y Vlahavas I. (2001) The GRT Planner, AI Magazine (2001), 63-65
- [Reitbauer et al., 2004] 57. Reitbauer, A., Battino, A., Karageorgos, A., Mehandjiev, P., Valckenaers, P., & Saint-Germain, B. (2004). The MaBE middleware: extending multi-agent systems to enable open business collaboration. International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services - BASYS04
- [REPAST 2009] <http://REPAST.sourceforge.net> (2009)
- [Richter y Hellenschmidt, 2004] Richter, K. & Hellenschmidt, M. (2004). Interacting with the Ambience: Multimodal Interaction and Ambient Intelligence. Position Paper to the W3C Workshop on Multimodal Interaction, 19-20 July
- [Riesbeck y Schank, 1989] Reisbeck C., Schank R. Inside Case-Based Reasoning. Lawrence Erlbaum Ass. Hillsdale, New Jersey, 1989.
- [Rodríguez-Aguilar, 2003] Juan Antonio Rodríguez-Aguilar, editor. On the design and construction of agent-mediated electronic institutions, Monografíes of IIIA No. 14, Barcelona. 2003.49

## Referencias

---

- [Rodríguez et al., 2007] I. Rodríguez, M. Salamo, M. Lopez-Sanchez, J. Cerquides, A. Puig, and C. Sierra. Completing the virtual analogy of real institutions via iobjects. In C CIA, 2007.
- [Rodríguez et al., 2008] Rodríguez, I.; A. Puig, M. Esteva, C. Sierra, A. Bogdanovych, S. Simoff. Intelligent Objects to Facilitate Human Participation in Virtual Institutions. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. ISBN:978-0-7695-3496-1, pp 196-199, 2008.
- [Rodríguez et al., 2009] Rodríguez S., Pérez-Lancho B., De Paz J.F., Bajo J. and Corchado J.M.: Ovamah: Multiagent-based Adaptive Virtual Organizations. 12th International Conference on Information Fusion, Seattle, Washington, USA. Julio 2009
- [Rosenblatt y Payton, 1989] Rosenblatt J.K. y Payton D.W. A fine-grained alternative to the subsumption architecture for mobile robot control. Proc of the IEEE Int. Conf. on Neural Networks, vol. 2, pp. 317—324. IEEE Press. 1989
- [Rosenschein y Zlotkin, 1994] Rosenschein, J.; y Zlotkin, G. (1994) Rules of Encounter - Designing Conventions for Automated Negotiation among Computers. MIT Press.
- [Rueda et al., 2002] Sonia Rueda, Alejandro J. García, and Guillermo R. Simari. Argument-based negotiation among bdi agents. Journal of Computer Science and Technology, 2(7):1-8, October 2002.
- [Russel y Norvig, 1995] Russell S. and Norvig P. (1995). Artificial Intelligence: A Modern Approach. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [Sacerdoti, 1975] Earl D. Sacerdoti. Planning in a hierarchy of abstraction spaces. Artificial Intelligence, 5(2):115--135, 1975
- [Sacerdoti, 1977] Earl D. Sacerdoti. A structure for Plans and Behaviour". American Elsevier, NY, 1977
- [Schank, 1982] Roger Schank. Dynamic Memory: A Theory of Learning in Computers and People (New York: Cambridge University Press, 1982)
- [Sam-Haroud y Faltings, 1996] Sam-Haroud D. y Faltings B. Consistency techniques for continuous constraints. Constraints, 1(1/2):85-118, September 1996
- [Samper, 2005] J. J. Samper. Ontologías para servicios web semánticos de información de tráfico: descripción y herramientas de explotación. PhD thesis, Universidad de Valencia - Departamento de Informática, 2005.
- [Sansores y Pavón, 2005] C. Sansores, J. Pavón. Simulación social basada en agentes. Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial, ISSN 1137-3601, Vol. 9, Nº. 25, 2005 , pags. 71-78
- [Searle, 1969] J. R. Searle. Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press, 1969.
- [Sen y Airiau, 2007] Sandip Sen, Stéphane Airiau: Emergence of Norms through Social Learning. IJCAI 2007: 1507-1512
- [Serrano y Ossowski, 2004] Serrano, J M., and S. Ossowski, "RICA-J -- A Dialogue-Driven Software Framework for the Implementation of Multiagent Systems", JISBD Taller en Desarrollo de Sistemas Multiagente (DESMA-2004), {M}álaga, pp. 48-61, Nov, 2004.

- [Schein, 1985] Edgar H. Schein. Organizational culture and leadership. Jossey-Bass. 1985. Edición: 3 - 2004 - 437 páginas
- [Schertler, 1998]. Schertler, W. "Virtual enterprises in tourism: folklore and facts: Conceptual challenges for academic research". in Information and Communication Technologies in Tourism. Buhalis, D., Min Tjoa, A. and Jafari J. editors. Springer. 1998.
- [Schultz, 1993] Schutz B.F. Geometrical Methods of Mathematical Physics. Cambridge U.P. 1993
- [Shoham, 1992] Y. Shoham and M. Tennenholtz. Emergent conventions in multi-agent systems. In Proceedings of Knowledge Representation and Reasoning, pages 225-231, 1992.
- [Shoham y Tennenholtz, 1997] Y. Shoham and M. Tennenholtz. On the emergence of social conventions: Modeling, analysis, and simulations. Artificial Intel ligenge, 94(1-2):139-166, 1997.
- [Smith, 1980] Smith, R.G.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. IEEE Transaction on Computers C-29(12), 1980, pp. 1104-1113.
- [Soley, 2005] Soley, R., the OMG Sta® Strategy Group: Model driven architecture. Object Management Group White Paper Draft 3.2 (2005)
- [Spalazzi, 2001] Spalazzi L. A Survey on Case-Based Planning. 2001
- [Spencer, 1896] Herbert Spencer. The Study of Sociology. New york. D. Appleton and company. 1896
- [Stanton y Walker, 2004] Stanton, E., y Walker, ed. Fundamentos de Marketing. 13a ed. 2004, McGraw Hill.668.
- [Stefik, 1981] Stefik, M. "Planning with constraints" (MOLGEN: Part 1). En Artificial Intelligence, 16(2): pag. 111-140, 1981
- [Sycara et al., 2003] Sycara, K., Paolucci, M., van Velsen, M., Giampapa, J.: The RETSINA MAS Infrastructure. Journal of Autonomous Agents and Multi-Agent Systems 7(1 & 2), Kluwer Academic Publishers, 2003.
- [Tambe, 1997] Tambe, M.: Towards Flexible Teamwork. Journal of Artificial Intelligence Research 7, 1997, pp. 83 - 124.
- [Taylor, 1991] Frederick Winslow Taylor. Principios y métodos de gestión científica, 1911
- [Terzopoulos , 1999] D. Terzopoulos, "Artificial Life for Computer Graphics", revista "Communications of theACM", agosto 1999, vol 42, num 8
- [Thimbleby, 2003] H. W. Thimbleby. The directed chinese postman problem. In journal of Software { Practice and Experience, 33(11):1081{1096, September 2003.
- [Vázquez y López de Ipiña, 2005] Vázquez, I., & López de Ipiña, D. (2005). Inteligencia Ambiental: la presencia invisible. Revista solo programadores No. 127, Revistas Profesionales 16-19.

## Referencias

---

- [Veloso y Rizzo, 1998] Manuela M. Veloso and Paola Rizzo, Mapping Planning Actions and Partially-Ordered Plans into Execution Knowledge.
- [Villatoro y Sabater-Mir, 2007]. D. Villatoro and J. Sabater-Mir. Norm Selection Through Simulation in a Resource-Gathering Society Proceedings of 21st European Simulation and Modelling Conference (ESM07)
- [Villatoro y Sabater-Mir, 2008a] D. Villatoro and J. Sabater-Mir. Categorizing Social Norms in a Simulated Resource Gathering Society. Proceedings of the AAAI Workshop on Coordination, Organizations, Institutions and Norms (COIN @ AAAI08)
- [Villatoro y Sabater-Mir, 2008b] D. Villatoro and J. Sabater-Mir. Mechanisms for Social Norms Support in Virtual Societies. Proceedings of the Fifth Conference of the European Social Simulation Association (ESSA08)
- [Villatoro y Sabater-Mir, 2008c]. D. Villatoro and J. Sabater-Mir. Towards the Group Formation through Social Norms. Proceedings of the Sixth European Workshop on Multi-Agent Systems (EUMAS08)
- [Von Martial, 1992] Von Martial, F. (1992): Co-ordinating Plans of Autonomous Agents. LNAI 610, Springer.
- [Wagner y Hollenbeck, 2004]. Wagner, J., y Hollenbeck, J. , Comportamiento Organizativo. 2004: Thomson.
- [Walker y Wooldridge, 1995] A. Walker and M. Wooldridge. Understanding the emergence of conventions in multi- agent systems. In ICMAS95,pages 384-389, 1995.
- [Wegner, 1997] Wegner, P. (1997) Why Interaction is more Powerful than Computing. Communications of the ACM 5(40), 80-91.
- [Weld, 1999] Weld, Daniel S.; "Recent Advances in AI Planning"; TR99, Department of Computer Science & Engineering, Universidad de Washington. En AI Magazine vol. 20, n.2, pág. 93-123, 1999.
- [West y Hubbold, 1998] West, A.,Hubbold, R.: Research challenges for systems supporting collaborative virtual environments. In Proceedings, Collaborative Virtual Environments 98, Manchester, p.11(1998)
- [Weyns et al., 2004] Weyns, D., Schelfhout, K., Holvoet, T., & Glorieux, O. (2004). Role based model for adaptive agents. Forth Symposium on Adaptive Agents and Multiagent Systems at the AISB04 Convention
- [Weiser, 1993] Weiser, M.: Ubiquitous Computing. IEEE Computer Hot Topics, October,1993, quoted in <http://www.ubiq.com/hypertext/weiser/UbiHome.html>.
- [Wiederhold, 1992] Wiederhold, G.: Mediators in the Architecture of Future Information Systems. IEEE Computer Magazine, March 1992.
- [Wooldridge, 1999] Wooldridge, M.: Intelligent Agents. Weiss, G. (Ed): Multi-agent Systems: a Modern Approach to Distributed Artificial Intelligence, MIT Press, 1999
- [Wooldridge, 2002] Wooldridge, M. (2002). An Introduction to MultiAgent Systems. Chichester, England, John Wiley & Sons, ISBN 047149691X.

- [Wooldridge y Jennings 1994] Wooldridge M. and Jennings N. R. (1994) 'Agent Theories, Architectures and Languages: A Survey', Intelligent Agents (ECAI-94 Workshop Proceedings on Agent Theories, Architectures and Languages, Amsterdam, Aug, 1994), pp 1-22, Wooldridge Michael J, and Jennings, Nicholas, (Eds), Springer Verlag, Berlin.
- [Wooldridge y Jennings, 1995] Wooldridge M. y Jennings N. R. (1995). Intelligent Agents: Theory and Practice. The Knowledge Engineering Review, vol. 10(2) pp. 115-152, 1995.
- [Yang, 1997] Yang, Qiang. "Intelligent Planning; A Descomposition and Abstraction Based Approach" Springer 252pag., 1997.
- [Zambonelli, 2002] Zambonelli, F.: Abstractions and Infrastructures for the Design and Development of Mobile Agent Organizations. In: Wooldridge, M., Weiss, G., Ciancarini P. (Eds.): Agent-Oriented Software Engineering II, LNCS 2222, Springer-Verlag, 2002, pp. 245 - 262.
- [Zambonelli y Parunak, 2002] F. Zambonelli and H. Parunak. 2002. From design to intention: Signs of a revolution. In Proc. 1st Int. Joint Conference on AAMAS: 455-456.
- [Zambonelli et al., 2001] Zambonelli F., Jennings, N., Wooldridge, M.: Organisational Abstractions for the Analysis and Design of Multi-Agent Systems. In: Ciancarini P., Wooldridge, M. (eds.): Agent
- [Zambonelli et al. 2003] F. Zambonelli, N. R. Jennings, and M. Wooldridge. Developing multiagent systems: The Gaia methodology. ACMTransactions on Software Engineering and Methodology, 12(3):317-370, July 2003.
- [Zambonelli et al. 2004] Zambonelli, F., Gleizes, M.-P., Mamei, M., & Tolksdorf, R. (2004). Spray computers: frontiers of self-organisation for pervasive computing. Workshop on Enabling Techonologies: Infrastructure for Collaborative Enterprises - WETICE04 (págs. 397-402). IEEE Computer Society



---

## Anexo A ANÁLISIS Y DISEÑO DEL CASO DE ESTUDIO

El desarrollo de SMA necesita de metodologías que permitan que el análisis y diseño de software basado en agentes pueda ser optimizado. Las primeras metodologías que surgieron se clasifican como *orientadas al agente*. En ellas no se describe la *organización* explícitamente. En este tipo de metodologías el diseñador se centra en las acciones individuales de los agentes y las estructuras sociales no se modelan, se forman con la interacción de los mismos. Ejemplos de estas metodologías son GAIA (1ª versión) [Zambonelli et al. 2003], TROPOS (1ª versión) [Bresciani et al., 2004] y PROMETHEUS [Padgham, 2002]. A continuación se exponen brevemente sus características principales:

- ❖ GAIA. Es una metodología que permite el análisis y diseño de sistemas multi-agentes concibiéndolos como organizaciones computacionales de agentes, los cuales juegan roles específicos y cooperan entre sí para alcanzar objetivos comunes. Consta únicamente de las fases de análisis y diseño. En la primera se definen los modelos de rol y de interacción. En la fase de diseño se definen los modelos de agente, servicio y de comunicación. Esta metodología asume un objetivo común entre todos los agentes del sistema.
- ❖ TROPOS. Esta metodología cubre todas las fases de desarrollo del software, desde los requisitos hasta la creación del código. En la fase de requisitos se modelan los actores principales del dominio el sistema y sus intenciones. En la fase de diseño de la arquitectura se descompone el sistema añadiendo nuevos actores, derivados del estilo arquitectónico.
- ❖ PROMETHEUS. Esta metodología también considera el ciclo completo del software para el diseño de sistemas multi-agente. Las fases de especificación y análisis están integradas y es ahí donde se identifican las percepciones y acciones del sistema, así como los objetivos y funciones del mismo. La fase de diseño se divide en arquitectura y en diseño detallado. La arquitectura se centra en la parte interna de cada agente y durante el diseño detallado se desarrolla esa arquitectura interna.

Todas estas metodologías ya han sido utilizadas por lo que han demostrado su funcionalidad, además se encuentran bien documentadas y sirven como base para nuevos desarrollos. Por contra, presentan algunas desventajas como la incapacidad para predecir comportamientos, establecer estrategias o aprovechar las características del dominio donde se integran. A partir del

estudio realizado se observa que todas estas metodologías comparten ciertos aspectos:

- ❖ Se establece un objetivo común entre los agentes
- ❖ Los agentes cooperan para conseguir el objetivo
- ❖ Se centran en la arquitectura de los agentes
- ❖ Son apropiadas para sistemas cerrados

Este último aspecto es importante. Los sistemas que se diseñaban eran generalmente cerrados y no permitían que agentes externos entraran en ellos. Pero recientemente, han surgido nuevas metodologías *orientadas a la organización* que permiten (parcialmente) el diseño de SMA permitiendo sistemas heterogéneos y abiertos. Estas metodologías *orientadas a la organización* permiten que agentes externos accedan a la funcionalidad del sistema, pero obligándoles a cumplir las *normas sociales* del sistema. En este tipo de metodologías, el diseñador se centra desde un principio en la organización del sistema, analizando el SMA desde una perspectiva global donde los objetivos de la organización representan una descripción a alto nivel de los propósitos de la sociedad. A su vez estas metodologías se pueden agrupar en dos tendencias, las que se centran en la estructura organizativa y las que se enfocan en las normas sociales.

Algunos ejemplos de estas metodologías son AGENT-GROUP-ROLE [Ferber et al., 2004], ROADMAP [Juan y Sterling, 2002] y la extensión de TROPOS [Bresciani et al., 2004], todas ellas de tipo estructura organizativa. Por otra parte, OPERA [Dignum 2004], CIVIL AGENT SOCIETIES [Dellarocas, 2000] y la extensión de GAIA [Zambonelli et al. 2003] son del tipo normas sociales. A continuación se exponen brevemente con sus características principales:

- ❖ AGENT-GROUP-ROLE. Esta metodología se basa en los conceptos de agente, grupo y rol donde dos agentes se comunican si pertenecen al mismo grupo. Las interacciones de los distintos agentes se detallan mediante redes de Petri. La consideración del grupo como estructura topológica hace que no se tengan en cuenta otras posibles relaciones estructurales entre los agentes como burocracia, matriciales, jerarquías, etc.
- ❖ ROADMAP. Esta metodología proviene y amplía la metodología GAIA y como ésta, también está constituida por dos fases: análisis y diseño. Esta metodología separa claramente ambas fases y abarca cualquier tipo de agente permitiéndose su integración con otras metodologías.
- ❖ Extensión TROPOS. Esta metodología adopta la metáfora organizativa y enfatiza el estudio explícito y la identificación de la estructura organizativa. Mediante la observación de las relaciones y estructuras humanas, sugiere posibles relaciones para agentes. Hace uso de la



teoría de las organizaciones y está compuesta de las mismas fases de desarrollo que TROPOS: requisitos iniciales, requisitos tardíos, diseño arquitectónico y diseño detallado.

- ❖ OPERA. Esta metodología realiza el diseño de sociedades de agentes desde una perspectiva organizativa concebida en tres niveles: modelo organizacional, modelo social y modelo de interacción. El modelo organizacional describe el comportamiento deseado o intencionado de la sociedad y su estructura general. En el modelo social se completa el modelo anterior mediante la asociación de roles y finalmente en el modelo de interacción se especifican los acuerdos de interacción entre actores.
- ❖ CIVIL AGENT SOCIETIES. Este entorno de trabajo se basa en las sociedades civiles humanas, donde se han desarrollado las instituciones para establecer y controlar la ejecución de las leyes. Esta metodología consta de varios servicios en los que se establecen las habilidades de los agentes respecto de las normas de la sociedad, se verifica que las interacciones entre los agentes son legales y se buscan las posibles excepciones para tratar de devolver la sociedad a un estado aceptable.
- ❖ Extensión GAIA. Esta extensión pretende diseñar sistemas abiertos empleando conceptos organizativos. En la fase de análisis se describe cómo se espera que trabaje la organización, identificando los objetivos y dividiendo en suborganizaciones, mientras que en la fase de diseño se describe qué tipo de organización se ajusta mejor.

Entre las metodologías más populares recientemente y que permiten el diseño de organizaciones están PASS [Cossentino et al., 2001], MOISE [Gateau et al., 2005], OperA [Dignum 2004] y GORMAS [Argente, 2008]. GORMAS (*Guidelines for Organization-based MultiAgent Systems*) es la metodología utilizada en este trabajo. Es una guía metodológica para el diseño de SMA abiertos desde la perspectiva de las organizaciones humanas. Sus características, junto con las fases que propone para cubrir el análisis, diseño de la estructura organizativa y diseño de la dinámica de la organización se verán en los siguientes subapartados. Es también esencial mencionar de nuevo las instituciones electrónicas [Esteva et al., 2001] como trabajo pionero en las organizaciones de agentes, en su propuesta de agentes estructurados en una organización (Institución Electrónica). Esta propuesta incluye mecanismos de regulación de comportamientos entre sus miembros pero donde la posibilidad de creación de nuevos grupos (organizaciones dinámicas) no está permitida. Se centra principalmente en el empleo de estructuras organizacionales durante el proceso de diseño y en la regulación de los agentes y es muy complicado crear nuevos grupos o sub-organizaciones dinámicamente en tiempo de ejecución.

El principal problema de la mayoría de estos trabajos es que no incluyen todas las fases de desarrollo de sistemas multi-agente abiertos basados en organizaciones. Es decir, no incorporan una guía desde la concepción de la idea del sistema hasta su implementación. Existen aproximaciones que emplean MDD (*Model Driven Development*) para el diseño de organizaciones y que intentan hacer esto. Estos enfoques emplean meta-modelos de organizaciones virtuales con conceptos y componentes en ciertos niveles de abstracción adecuados para poder implementar los sistemas en diferentes plataformas. Algunas de las metodologías que integran las técnicas de MDD en el diseño de SMA son: INGENIAS [García-Magario et al., 2009] TROPOS [Perini y Susi, 2006] y PIN4AGENT [Hahn et al., 2008]. Estos trabajos aplican MDD al proceso de modelado de agentes, pero el mayor problema que se encuentra en ellos es que no consideran el desarrollo de organizaciones virtuales. Por esta razón, para el diseño del caso de estudio se han seguido los trabajos de [Agüero et al., 2009a] [Agüero et al., 2009b] [Agüero et al., 2010] en los que se aplican el enfoque MDD en metodologías orientadas a organizaciones. Es posible diseñar la organización de forma unificada, intuitiva y con un alto nivel de abstracción. Mediante esta aproximación se facilita y se simplifica el proceso de diseño de SMA orientado a organizaciones obteniendo modelos de organizaciones virtuales que pueden ser implementados en diferentes plataformas. La idea fundamental es la de crear modelos diferentes para distintos niveles de abstracción, combinándolos para llegar a una implementación. Estos meta-modelos definen la estructura, la semántica y las restricciones para una familia de modelos. Los tipos de modelos que se diferencian son los siguientes:

- ❖ *Computational Independent Model (CIM)*. Define los conceptos generales independientemente si van a ser implementados en una máquina u otra.
- ❖ *Platform Independent Model (PIM)*. Define las funcionalidades del sistema independientemente si van a ser implementadas en una plataforma u otra.
- ❖ *Platform Specific Model (PSM)*. Combina el modelo PIM con los detalles específicos de la plataforma donde se va a implementar la organización.

En el siguiente esquema puede observarse la relación entre los diferentes modelos. Como se ha indicado, se ha utilizado GORMAS [Argente, 2008] como metodología de diseño y THOMAS [Carrascosa et al., 2009] [Giret et al., 2009] como plataforma final de desarrollo. El MDD propuesto en [Agüero et al., 2009a] [Agüero et al., 2009b] [Agüero et al., 2010] facilita el conjunto de reglas necesarias para las transformaciones entre los modelos. Así podrá distinguirse entre transformaciones horizontales (PIM-to-PIM) y verticales (PIM-to-PSM).

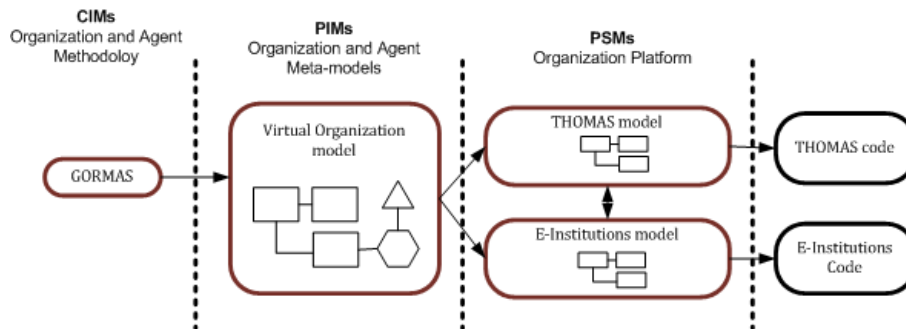


Figura 27. Meta-modelos en MDD

El meta-modelo de organización describe los componentes que serán incluidos en la misma, definiendo los componentes principales y los conceptos básicos empleados para modelar la organización, los cuales se resumen en 5 meta-modelos: estructural, funcional, normativo, agente y entorno [Agüero et al., 2009b] [Argente, 2008].

Primero se elaboran estos meta-modelos para la caracterización, de forma más abstracta, de la organización. Una vez estos meta-modelos están definidos, empieza el proceso MDD para la transformación en la organización concreta. El proceso de desarrollo de la organización virtual usando MDD está formado por un conjunto de pasos o fases. Estas fases, junto con las herramientas requeridas para cada uno de ellas son las siguientes:

1. Creación del modelo. Se crean los diagramas que representan la organización modelando las unidades, roles, tareas, etc. Para llevar a cabo este paso se emplean las herramientas: Eclipse IDE y los plugins<sup>39</sup> EMF, EMFGormas, Ecore, GMF y GEF, que permiten modelar la organización de manera gráfica. La notación empleada en esta fase es la empleada en GORMAS [Argente, 2008].
2. Selección de la plataforma. En nuestro caso, la organización será implementada sobre la plataforma THOMAS. En este paso es necesario aplicar una transformación modelo-modelo (PIM-to-PSM). Para la consecución de este paso se usará Eclipse IDE junto con el plugin ATL<sup>40</sup> que incorpora las reglas de transformación entre modelos necesarias.
3. Generación del sistema. Se aplicará una transformación PSM-to-code. En este caso se emplea MOFScript<sup>41</sup>, otro plugin de Eclipse que utiliza

<sup>39</sup> <http://users.dsic.upv.es/grupos/ia/sma/tools/EMFGormas/index.html> [Último acceso 23/12/2009]

<sup>40</sup> <http://dev.eclipse.org/viewcvs/indextech.cgi/gmt-home/subprojects/ATL/index.html> [Último acceso 23/12/2009]

<sup>41</sup> <http://www.eclipse.org/gmt/mofscript/> [Último acceso 23/12/2009]

plantillas para hacer el proceso de transformación. En los trabajos de [Agüero et al., 2010] se han desarrollado las plantillas necesarias para la generación de código para las plataformas THOMAS [Carrascosa et al., 2009] [Giret et al., 2009] e E-Institutions [Esteva et al., 2001]. La idea básica es que se proporcionan los ficheros XML (eXtensible Markup Language) que describen los componentes y las relaciones del meta-modelo para después generar otro XML que contiene la especificación de THOMAS que será la aplicación ejecutable.

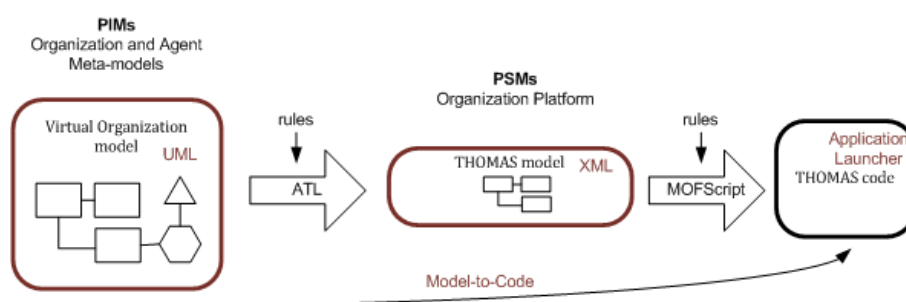


Figura 28. Procesos entre metamodelos

Los siguientes sub-apartados mostrarán de manera más detallada los pasos que se llevaron a cabo para la creación del modelo. Como se ha indicado, la metodología utilizada es GORMAS, por lo que en primer lugar se hará una breve descripción de la misma.

## A.1 GORMAS

GORMAS es una guía metodológica para el diseño de sistemas multi-agentes abiertos desde la perspectiva de las organizaciones humanas [Argente, 2008]. Esta guía se basa en la Teoría de Organización y en el Modelo de Organización [Argente, 2008]. Esta teoría aborda las propiedades de las organizaciones humanas y sus aspectos de diseño, mientras que el Modelo de Organización describe los principales aspectos de las organizaciones: estructura, funcionalidad, normalización, dinamicidad y entorno. Este modelo consta de un conjunto de meta-modelos que emplean fundamentalmente los conceptos de unidad organizativa, servicio, norma y entorno. Además, contiene un conjunto de patrones de diseño con los que facilitar el modelado de la estructura de la organización.

GORMAS consta de un conjunto de fases que cubren el análisis, el diseño de la estructura organizativa y el diseño de la dinámica de la organización. Con estas fases se especifica cuáles son los servicios que ofrece la organización, cuál es su estructura interna y qué normas rigen su comportamiento. Las fases son las siguientes:

- ❖ Fase A. Misión: se realiza un análisis de la motivación que se persigue al definir la organización o sistema, es decir, del porqué de dicha organización o para qué se crea; de los resultados que, en conjunto, se esperan conseguir; y del entorno en el que existe, detallando los productos y/o servicios a ofrecer, los grupos de interés y su localización.
- ❖ Fase B. Tareas y procesos: se analizan con mayor detalle los servicios a ofrecer en el sistema, sus requisitos y los procesos que conllevan. Se detallan también las tareas y objetivos asociados a dichos servicios.
- ❖ Fase C. Dimensiones organizativas: se analizan las dimensiones de la organización (departamentalización, especialización, sistema decisor, formalización, coordinación), que imponen ciertos requisitos sobre los tipos de trabajo, así como sobre la diversidad e interdependencia de las tareas a realizar.
- ❖ Fase D. Estructura organizativa: se determina y selecciona la estructura organizativa más adecuada para la organización, en función de sus dimensiones. Se hace uso de modelos organizativos para especificar los roles, interacciones y normas relacionados con la propia estructura.
- ❖ Fase E. Procesos de información-decisión: para cada servicio identificado, se detallan las interacciones (flujos de información y de adopción de decisiones) necesarios para llevar a cabo el servicio. Además, se definen los contratos de calidad de servicio a los que se comprometen los proveedores y consumidores cuando éste se lleva a cabo.
- ❖ Fase F. Dinamicidad del sistema abierto: se establece la funcionalidad ofrecida como sistema abierto, que incluye los servicios que se deben publicitar y las políticas de adquisición y liberación de roles. Además, se diseñan los agentes propios del sistema.
- ❖ Fase G. Sistemas de medición, evaluación y control : se cuantifican o evalúan las tareas y actividades y se establecen mecanismos para determinar si los objetivos del sistema se cumplen. Asimismo, se revisan las normas de la organización para especificar quiénes se encargan de ellas y las supervisan.
- ❖ Fase H. Sistemas de recompensas: se determina el sistema de incentivos, para recompensar a los miembros que avancen en dirección de los intereses de la organización. También se analizan los sistemas de sanción para aquellos miembros que no cumplan con las normas dadas.

## Anexos

---

La secuencia-guía propuesta permite ser integrada en un proceso de desarrollo de software completo, que comprende las fases de análisis, diseño, implementación, instalación y mantenimiento del SMA.

---

## A.2 ANÁLISIS DEL SISTEMA

---

El caso de estudio presenta una sociedad orientada a ofrecer servicios de guiado para turistas. Dicha sociedad se implementa utilizando el modelo de planificación propuesto en el capítulo 5 y se representa en un mundo virtual en el cual estarán disponibles los inventarios de un conjunto documental de patrimonio cultural. No se trata de un sistema de guiado típico. En los sistemas de guiado clásicos cada guía es asignado a un grupo de turistas y visitantes y realiza una ruta turística asignada en la empresa para este grupo. En nuestro caso de estudio el sistema es mucho más abierto. La planificación inicial, tanto la asignación de grupos de turistas (y visitantes) a guías, como la ruta a seguir, puede variar. En función de los guías disponibles, de las preferencias y situación de los turistas y de los eventos disponibles para realizar una ruta, se realiza una planificación dinámica para un momento concreto; lo que puede provocar que la sociedad de agentes tenga que adaptarse a las nuevas necesidades. En la elaboración de este caso de estudio se ha limitado el tipo de turistas a visitantes que no varían sus datos a lo largo del tiempo de ejecución de las pruebas. Para evaluar la adaptación del modelo en la replanificación y reorganización se han aprovechado las variaciones de los Guías.

Para el análisis de los requisitos del sistema se tienen en cuenta la *Fase A. Misión* y la *Fase B. Tareas y Procesos* de GORMAS.

---

### A.2.1 Fase A. Misión

---

En esta fase se va a definir el porqué del sistema que se pretende desarrollar, la misión de la organización. La misión de una empresa se define como la satisfacción de determinadas necesidades o deseos de ciertos grupos de interés que, en conjunto, forman sus mercados[Casillas 2004]

Para [Ferrel y Hirt, 2004] la misión de una organización es su propósito general. Responde a la pregunta ¿qué se supone que hace la organización?. En nuestro contexto la misión del sistema es administrar un conjunto de enclaves turísticos que serán visitados por grupos de turistas y sobre los que tendremos información que aprovecharán los guías de nuestra organización para realizar sus rutas. Por tanto, lo que se pretende es conseguir la mayor satisfacción por parte de los turistas con el mínimo de recursos empleados por la organización.

Para Stanton, Etzel y Walker, autores del libro «Fundamentos de Marketing», la misión de una organización "enuncia a qué clientes sirve, qué necesidades satisface y qué tipos de productos ofrece. Por su parte, una declaración de misión indica, en términos generales, los límites de las actividades de la organización" [Stanton y Walker, 2004].

Por tanto, definir la misión de un sistema incluye describir el resultado que produce el sistema, identificar los principales grupos de interés a los que va dirigido el sistema e identificar el entorno en el que se va a desarrollar. Estos elementos se detallarán en la misión organizativa, los grupos de interés y las condiciones del entorno.

En la tabla que se muestra a continuación se observan los resultados (productos o servicios) que ofrece el sistema, el tipo de entorno sobre el que se trabaja, así como los grupos de interés a los que afecta. También se justifica la razón de ser del sistema, el cual pretende facilitar la integración de los agentes y los servicios web.

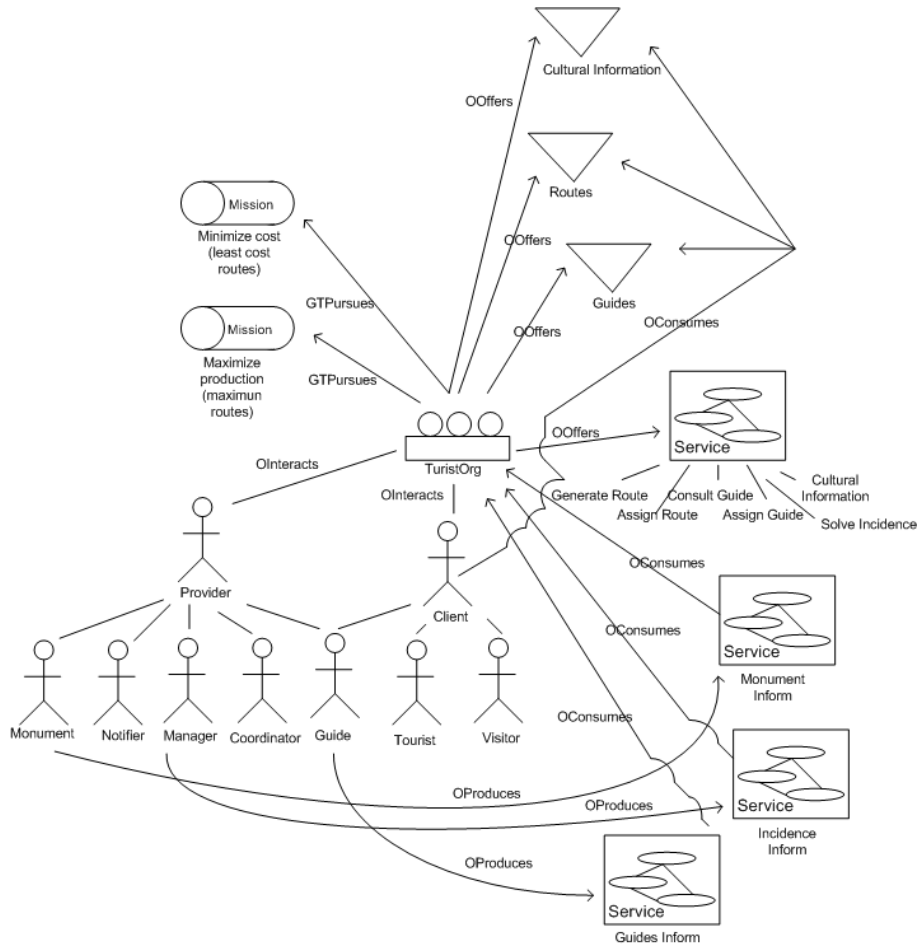
**Tabla 6. Documento A.1 - Misión Organizativa del sistema**

<i>A.1 Misión Organizativa</i>	
<b>Nombre:</b>	TouristOrg
<b>Dominio:</b>	turismo
<b>Resultados:</b>	<ul style="list-style-type: none"> <li>❖ <i>Producto: rutas. Finalidad:</i> itinerario asociado a un grupo de turistas y guía.</li> <li>❖ <i>Producto: información cultural. Finalidad:</i> Datos asociados a un monumento del enclave turístico.</li> <li>❖ <i>Producto: guías. Finalidad:</i> Información o perfil asociado a un guía y que lo especializa en un campo concreto.</li> <li>❖ <i>Servicio: generar ruta. Finalidad:</i> generar un itinerario concreto para un guía asignado a un grupo de turistas.</li> <li>❖ <i>Servicio: asignar ruta. Finalidad:</i> fijar rutas a guías y/o turistas.</li> <li>❖ <i>Servicio: consultar guía. Finalidad:</i> examinar el perfil de un guía.</li> <li>❖ <i>Servicio: asignar guía. Finalidad:</i> fijar guías a grupos de visitantes y/o turistas.</li> <li>❖ <i>Servicio: consultar información cultural. Finalidad:</i> examinar los datos sobre un monumento.</li> <li>❖ <i>Servicio: resolver incidencia. Finalidad:</i> solucionar un percance en el desarrollo de una ruta turística.</li> </ul>
<b>Grupos de interés:</b>	<ul style="list-style-type: none"> <li>❖ Cliente: solicita información al sistema para confeccionar su ruta turística. <ul style="list-style-type: none"> <li>○ Turista</li> <li>○ Visitante</li> <li>○ Guía</li> </ul> </li> <li>❖ Proveedor: proporciona información sobre rutas, información cultural, incidencias o perfiles. <ul style="list-style-type: none"> <li>○ Monumento: facilita información sobre los monumentos del enclave turístico.</li> <li>○ Notificador: facilita información sobre incidencias.</li> <li>○ Manager: facilita información sobre soluciones a incidencias e</li> </ul> </li> </ul>



<p>e históricos.</p> <ul style="list-style-type: none"> <li>○ Coordinador: facilita información sobre las rutas a realizar por cada guía.</li> <li>○ Guía: facilita información sobre la planificación de la ruta de cada guía.</li> </ul>
<p><b>Tipo de entorno:</b> virtual y distribuido</p>
<p><b>Justificación:</b> sistema abierto que facilita un punto de encuentro entre entidades demandantes y oferentes, de forma controlada sobre los servicios y productos que ofrecen y generan.</p> <ul style="list-style-type: none"> <li>• Sistemas similares: servicios web para la industria turística.</li> <li>• Ventajas: facilitar la adaptación del sistema.</li> <li>• Desventajas: mayor complejidad al ser un sistema abierto.</li> <li>• Singularidades: potenciar la heterogeneidad de los miembros del sistema y la capacidad de reorganización y adaptación.</li> </ul>

En la Tabla 6 se muestran los productos y servicios que ofrece el sistema, el tipo de entorno, los objetivos globales que persigue (misión y justificación), los grupos de interés a los que afecta y la información que consume. Siguiendo las pautas indicadas en la *Fase A. Misión*, se instancia la vista funcional (misión) del modelo de organización la cual representa todos estos parámetros.



**Figura 29. Diagrama del modelo de organizaci3n (vista funcional). Misi3n del sistema.**

El sistema ofrece como producto principal las rutas turísticas, que ser3n requeridas por los guías o los propios turistas. Tambi3n ofrece como productos la informaci3n cultural sobre los monumentos del sistema. El servicio encargado de informar sobre las características de cada monumento ser3 tarea del agente monumento.

Las guías tambi3n se ofrecen como "producto" ya que contendr3n cierta informaci3n personal, es decir su perfil, generada por ellos pero que luego podr3 ser utilizada y consultada por otros miembros de la organizaci3n pasar realizar asignaciones. En este caso, un guía podr3:

- ❖ Ser cliente cuando quiera obtener del sistema informaci3n sobre las rutas a seguir o sobre los monumentos que se les ha asignado.

- ❖ Ser proveedores cuando ofrezcan los planes de las rutas que han generado en el sistema o cuando den información sobre su perfil "Servicio Información Guías".

El sistema ofrecerá los servicios necesarios para generar y asignar rutas a los guías, consultar la información de los guías (perfil), asignar guías a turistas y consultar información sobre monumentos (información cultural).

También podrá resolver incidentes (como por ejemplo que un monumento esté o no disponible para generar rutas nuevas). Estos incidentes vendrán proporcionados por el Manager, que será el encargado de informar de todo lo que ocurre en y para el sistema.

La misión de la organización será, por un lado minimizar el coste del sistema de guiado, intentando generar mejores rutas con menos recursos (tiempo, dinero, etc.); y por otro lado, maximizar la producción, es decir, generar más beneficios haciendo cuantas más rutas mejor. Los dos objetivos son complementarios y proporcionales, es decir, a mínimo coste con más rutas implica mayor producción.

La siguiente tabla muestra un análisis más detallado de los *stakeholders* o grupos de interés que representan a aquellos actores que se verán afectados tanto por el éxito o el fracaso de la organización [Wagner y Hollenbeck, 2004]. Los grupos de interés se pueden dividir en primarios o secundarios dependiendo de si resultan imprescindibles para el sistema o si se puede prescindir de ellos.

Se identifican dos grupos de interés principales: los demandantes, que pueden ser turistas o guías que buscan una ruta, y los oferentes, que son los miembros de la organización que pueden generar y planificar rutas o que pueden ofertar algún tipo de servicio informativo. Dentro del grupo de los oferentes podemos distinguir cuatro subgrupos: monumentos, managers, notificadores y coordinadores.

Tabla 7. Documento A.2 Grupos de Interés del sistema

<i>A.2. Grupos de interés</i>		
<b>Stakeholder</b>		
	<b>Demandante</b> (Turista, Vistante, Guía)	<b>Oferente</b> (monumentos, managers, notificadores y coordinadores.)
Beneficiario	Primario	Primario
Tipo	Cliente	Proveedor
Objetivos	Encontrar la mejor ruta turística de acuerdo a sus necesidades	Maximizar número de rutas

Anexos

Requiere	Ruta turística con diferentes tareas	Realizar rutas
Proporciona	Retroalimentación sobre las rutas realizadas	Rutas turísticas
Frecuencia	Frecuente	Frecuente
Beneficios	Obtener una ruta turística económica y de garantía	Incrementa número de rutas realizadas y número de clientes turistas
Poder de decisión	Si (solo guías: proporcionar nuevas rutas)	Si (proporcionar nuevas rutas)
Influencia sobre sus intereses	Si (ofrecerle nuevas rutas)	

En la siguiente tabla (Tabla 8) se muestra el análisis de las condiciones de entorno. El entorno se refiere a todos aquellos factores externos a la organización que tienen una influencia significativa sobre ella.

Existen ciertas condiciones del entorno que afectan en gran medida al rendimiento y efectividad de la organización. Estas condiciones son la tasa de cambio, complejidad, incertidumbre, receptividad y diversidad [Wagner y Hollenbeck, 2004]. La tasa de cambio indica el grado en el que las condiciones del entorno cambian de forma impredecible. En este caso se considera un entorno dinámico, ya que, la demanda de los productos/servicios no es predecible y pueden aparecer nuevos elementos. La complejidad indica el número y relación entre sí de los elementos del entorno que afectan a la organización. Se considera que la complejidad es simple, debido a que está formada por un número pequeño de elementos. La incertidumbre refleja la falta de información sobre los factores, actividades y acontecimientos del entorno. Se puede decir que la incertidumbre del entorno de estudio es alta, ya que es un entorno dinámico. La receptividad indica el grado con que el entorno facilita o dificulta el progreso de una organización hacia sus objetivos. Se considera que el entorno es hostil puesto que la información es proporcionada por los operadores que puede que no estén siempre activos. La diversidad representa el número de sectores o escenarios distintos del entorno. Se dice que es un entorno uniforme ya que los servicios proporcionados están muy relacionados.

**Tabla 8. Documento A.3- Condiciones de Entorno del sistema**

<i>A.3. Condiciones del Entorno</i>		
<b>Condición</b>	<b>Valor</b>	<b>Justificación</b>
Tasa de Cambio	Dinámico	Podrán aparecer nuevos tipos de guías, y con ellos nuevos perfiles de

		especialización en rutas.
Complejidad	Simple	Existen pocos elementos distintos. Los productos a generar (rutas) tienen características similares.
Incertidumbre	Alta	Entorno dinámico y cambiante.
Receptividad	Hostil	Se requiere información proporcionada por proveedores que no siempre tienen que estar disponibles.
Diversidad	Uniforme	Servicios proporcionados son dependientes entre sí.

#### A.2.2 Fase B. Tareas y procesos

En esta fase se analiza la tecnología que requiere el sistema, teniendo en cuenta las características de los productos/servicios, los objetivos asociados a los servicios y las tareas y procesos a realizar, así como los objetivos de la organización.

La tecnología se encarga de los procesos de transformación de los recursos básicos en bienes o servicios finales; de la forma en la que los miembros de la organización realizan sus tareas; de la distribución de los productos y servicios; y de cualquier actividad necesaria en la organización. Además, una organización adoptará cierta estructura dependiendo de la tecnología empleada [Hodge y Gales, 2003]. Desde el punto de vista de los SMA, la tecnología comprende a los recursos, aplicaciones, conocimientos requeridos por los agentes y a los procesos y tareas necesarios para llevar a cabo los servicios que ofrece la organización. Existen tres niveles de tecnología a considerar: tecnología esencial o tecnología a nivel organizativo, que hace referencia a toda la organización, tecnología departamental o de unidad de trabajo, que contempla la diversidad y complejidad de las distintas tareas organizativas; y relaciones interdependientes, originadas como resultado del flujo de trabajo entre las unidades [Hodge y Gales, 2003].

Tabla 9. B.1. Tecnología esencial del sistema

<i>B.1 Tecnología esencial</i>	
<b>Tecnología de producto:</b> <i>producción en lotes pequeños</i>	
<ul style="list-style-type: none"> <li>• En masa: se elaboran paquetes de tareas para rutas turísticas con independencia del cliente final</li> <li>• En lotes pequeños: se confeccionan paquetes exclusivos para los clientes, según los perfiles de los guías.</li> </ul>	
<b>Tecnología de servicio:</b> <i>vinculación prolongada.</i> Existe un orden preestablecido de ejecución de los servicios del sistema: (i) asignación de guías, (ii) búsqueda del guía, (iii) asignación de tareas en una ruta turística, (iv) realización de la ruta.	

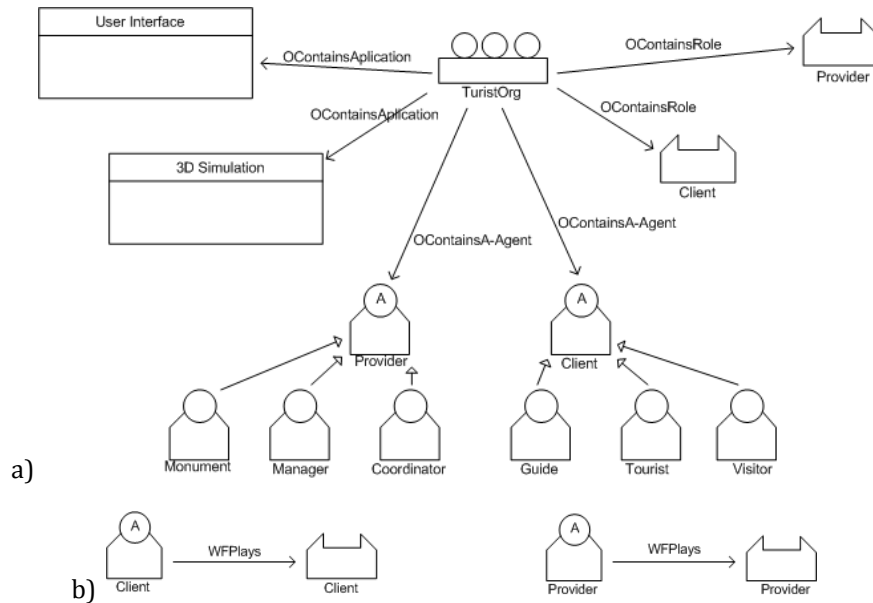
En primer lugar, la *tecnología esencial* se determina el modo en el que se deben producir los productos o servicios, en función de la demanda y las necesidades del cliente. Se puede dividir la producción de los productos en cuatro tipos: producción orientada a lotes pequeños, producción en masa, producción de proceso continuo y producción flexible.

En este caso se considera que la producción está orientada en lotes pequeños, donde los productos se ajustan a las necesidades de los turistas y visitantes, los guías se coordinan de forma exclusiva para un grupo de turistas determinado bajo demanda. Por tanto, los turistas y visitantes se consideran como miembros del sistema y se definen varios A-Agente que los representen.

Desde el punto de vista de los servicios, su tecnología se puede dividir en tres tipos: de vinculación prolongada, intensiva o de mediación. Teniendo en cuenta la tecnología de servicios se utiliza una tecnología de vinculación prolongada. Existe un orden establecido para la ejecución de los servicios del sistema: asignación de guías, búsqueda del guía, asignación de tareas en una ruta turística, realización de la ruta.

Al ser un sistema de producción en lotes pequeños, los clientes (turistas) especifican las características del producto a través de una interfaz de usuario de una aplicación. Esta aplicación pone en contacto a los turistas con guías turísticos, unos solicitan rutas que realizar y los otros proporcionan información sobre los monumentos que se incluirán en la ruta de ese turista.

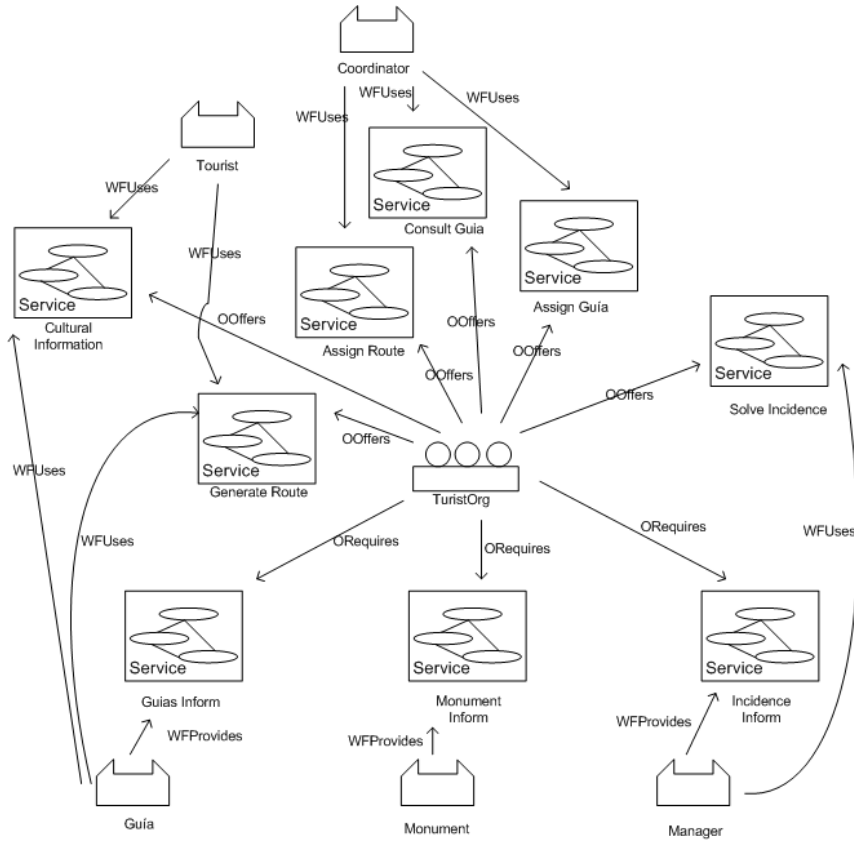
En la Figura 30 se muestra la especificación de la vista estructural para el sistema de guiado en el que se representa la interfaz de usuario. En este diagrama aparece también la entidad A-Agente que representa al cliente, la cual juega el rol cliente dentro de la unidad y la entidad A-Agente que representa al proveedor. También se representa una forma de interacción que será propia del sistema, la simulación 3D, que proporcionará el conducto de comunicación con la organización en los estudios simulados. La relación con la unidad organizativa se realiza mediante OcontieneAAgente, OcontieneRol y OContieneAplicación.



**Figura 30. Diagrama del modelo de organización del sistema. a) Vista estructural. b) vista funcional, funcionalidad interna**

Como se ha indicado, la tecnología de servicio que se requiere es una tecnología de vinculación prolongada, ya que el orden de invocación de los servicios es muy importante: asignación de guías, búsqueda del guía, asignación de tareas en una ruta turística, realización de la ruta.

En el diagrama de la Figura 31 se muestra la vista funcional (funcionalidad externa) del modelo de organización asociado a TouristOrg, en el que todos los servicios conectados entre sí están asociados a los roles del sistema con las relaciones WFUtiliza y WFProporciona. También requerirá de ciertos servicios que le proporcionen información. En este caso se representa la información que puede ser obtenida mediante la invocación a servicios proporcionados por agentes que no tienen por qué ser de la organización pero que pueden tomar los roles de Monumento, Guía y Manager para proveer esos datos a los agentes de la organización.



**Figura 31. Diagrama del modelo de organización, vista funcional, (funcionalidad externa)**

En segundo lugar, en el análisis de la tecnología de unidad de trabajo se detallan cada uno de los servicios identificados en la vista funcional anterior y se crea el correspondiente modelo de actividad. En el modelo de actividad se asocia una entidad de PerfilServicio (relación WFESpecifica) en la que se detallan las entradas y salidas que requiere.

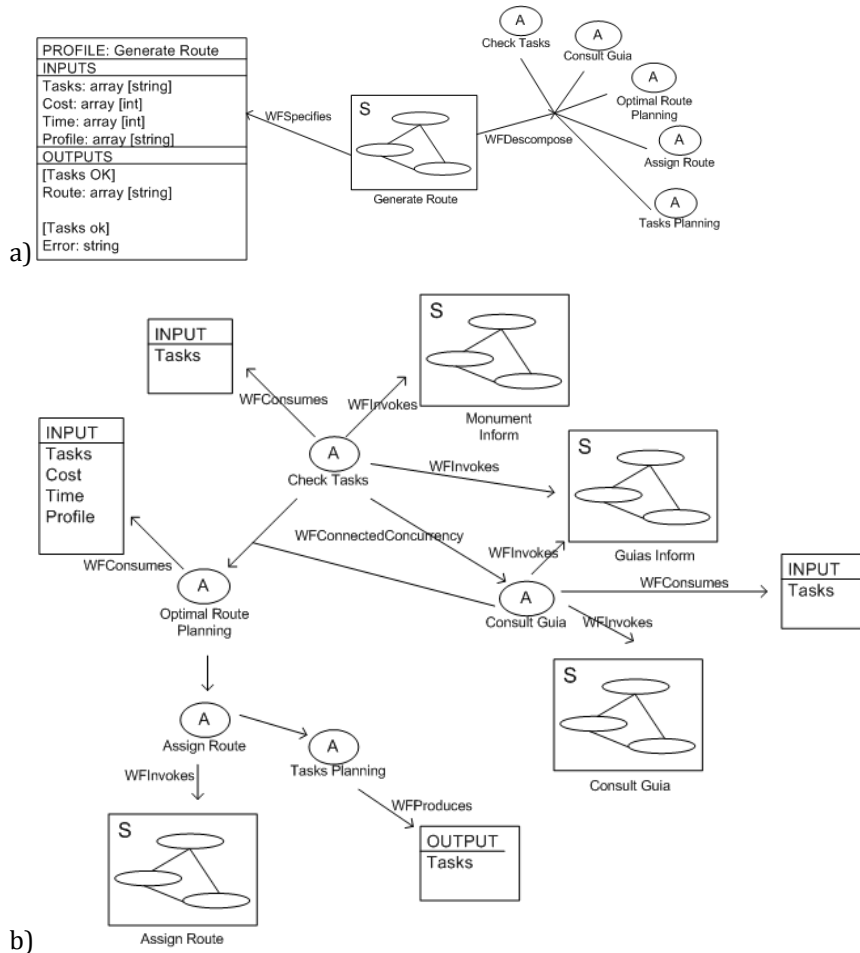
Se realiza una primera descomposición del servicio en una serie de tareas que precisa para su funcionamiento (entidad A-Tarea y relación WFDescompone). Por último, se determina si el servicio requiere de ciertos recursos y cuáles son los productos obtenidos (relaciones WFConsume y WFProduce).

La siguiente tabla muestra la descripción de uno de los servicios esenciales del sistema, *Generar Ruta* y la Figura 32 muestra el diagrama del modelo de actividad asociado al servicio y las A-Tareas en las que se descompone.

**Tabla 10. Documento B.2 Tecnología de Unidad de Trabajo. Descripción de las características del servicio.**



<i>B.2. Tecnología de Unidad de Trabajo</i>	
<b>Servicio</b>	<i>Generar Ruta</i>
Descripción	Realiza un itinerario dentro del enclave turístico para un guía (o turista) determinado (ej: (i) Visitar catedral (ii) Paseo caso antiguo (iii) Comida típica (iv) Regreso a estación)
<b>Consumidor</b>	Guía o Turista
Objetivo	Obtener una ruta de guiado turístico
Beneficio	Garantizar que realiza la ruta óptima
<b>Productor</b>	Coordinador
Objetivo	Planificar ruta turística
Beneficio	Asegurar mínimo coste para la ruta con máximos puntos en la ruta (acciones o tareas realizadas)
<b>Perfil del servicio</b>	
<i>Entradas</i>	<i>tareas + coste + tiempo + perfil guía</i>
	-tareas: conjunto de actividades que se desea realizar
	-coste: coste asociado a esas tareas
	-tiempo: tiempo asociado a cada tarea
	-perfil guía: especialización del guía para ciertas tareas
<i>Precondiciones</i>	Las tareas son válidas (pueden ser llevadas a cabo por los guías)
<i>Salidas</i>	Itinerario a seguir y coste asociado
<i>Postcondiciones</i>	Establecer como ocupado un guía cuando está realizando una ruta. Almacenar el éxito o fracaso del itinerario llevado a cabo por ese guía junto con el coste asociado.
<b>Funcionalidad</b>	
<i>Tareas</i>	<i>Comprobar tareas + Consultar guía + planificar ruta óptima + asignar ruta + planificar tareas</i>
	-Comprobar tareas: determinar que las tareas o acciones de la ruta pueden ser llevadas a cabo
	-Consultar guía: consultar perfil de guía
	-Planificar ruta: Generar el itinerario general que llevarán los guías
	-Asignar ruta: fijar cada tarea de la ruta al guía apropiado
	-Planificar tareas: obtener el mejor itinerario de cada guía (cada uno con sus tareas).
<i>Recursos</i>	Repositorio de tareas de ruta
<i>Proveedor</i>	TouristOrg (sistema de guiado)
<i>Productos</i>	Rutas



**Figura 32. Diagrama del modelo de actividad del servicio Asignar Guía. a) Perfil del servicio b) Relaciones entre las A-Tareas que lo componen**

El acceso a los servicios se indica a través de las relaciones WFInvoca de los diagramas de modelo de actividad. Por su parte, los productos que se generan (rutas) se almacenan en una base de datos o repositorio (Repositorio de rutas) para facilitar su gestión. Hay que recordar que el sistema tiene como pilar la utilización de un modelo CBR para conseguir las rutas óptimas y que esto es un punto importante en el análisis del sistema. En la Figura 33 se muestra el modelo de entorno de la unidad organizativa TouristOrg. El sistema es percibido por la entidad agente simulador quien se encarga de actuar en el sistema en nombre del usuario.

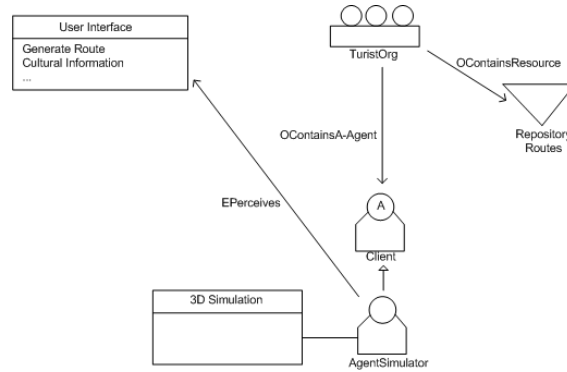


Figura 33. Diagrama del modelo de entorno del sistema

Por último, además de la misión, existen otras metas u objetivos de la organización que se derivan de ella: los objetivos funcionales y los objetivos operativos [Wagner y Hollenbeck, 2004]. Los objetivos funcionales representan las acciones específicas de las divisiones, departamentos o unidades de la organización, así como los resultados que se espera que consigan. Los objetivos operativos son los resultados específicos y medibles que se espera que consigan los miembros de una unidad. En la Figura 34 se muestra el diagrama de *modelo de actividad* para los objetivos funcionales.

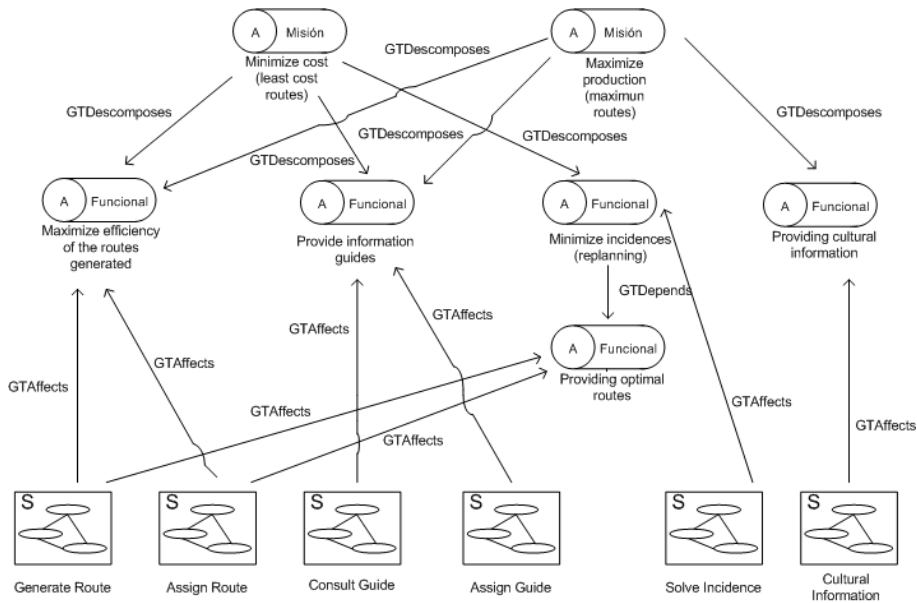


Figura 34. Diagrama del modelo de actividad. Relación entre servicios y objetivos funcionales.

### *A.3 DISEÑO DE LA ESTRUCTURA DE LA ORGANIZACIÓN*

---

En esta fase, después del análisis de las propiedades de la organización, se realiza un diseño de las dimensiones organizativas que imponen ciertos requisitos sobre los tipos de trabajo a realizar, sobre la estructura del sistema y sobre las interdependencias entre las tareas. Primero se analizarán las dimensiones (*Fase C*) que tratan de definir las características intrínsecas de la organización para posteriormente definir la estructura organizativa a adoptar (*Fase D*).

---

#### *A.3.1 Fase C. Dimensiones Organizativas*

---

En la Tabla 11 se muestran los valores obtenidos para las dimensiones organizativas del sistema del caso de estudio.

La fase de análisis de las dimensiones organizativas se va a dividir en dos apartados:

- ❖ Asignar tareas
  - Agrupación de las tareas en unidades
  - Asignación a los diferentes miembros de la organización (especialización).
- ❖ Identificar restricciones.
  - Identificar los mecanismos de coordinación
  - Definir la normalización.

Para **agrupar las tareas** se tiene en cuenta la dimensión denominada departamentalización, que consiste en agrupar los trabajos para coordinar tareas comunes. Dicha agrupación se realiza de dos modos: por funciones o divisional [Wagner y Hollenbeck, 2004].

En este caso se considera una agrupación funcional. Se agrupa en base a los conocimientos, habilidades y funciones que se realizan. A la hora de realizar la elección de la agrupación se ha tenido en cuenta la diversidad del entorno, el entorno a considerar es uniforme luego resulta preferible una **departamentalización funcional**.

Al analizar la **especialización** se tiene en cuenta la división del trabajo. Esto indica el grado en el que las tareas de la organización se dividen en cometidos separados. Para analizar la especificación se tienen en cuenta los servicios proporcionados por cada rol identificado (vista funcional), se analizan las A-

Tareas asignadas a cada uno de esos servicios en el modelo de actividad y se determina el tipo de interdependencia existente entre las tareas. Los tipos de interdependencias posibles son: recursos compartidos, secuencial, recíproca e integral [Wagner y Hollenbeck, 2004]. En nuestro estudio se identifican las interdependencias de recursos compartidos y secuenciales.

Se considera que el sistema tiene **especialización horizontal y ampliación vertical**. Los roles están muy especializados, por tanto, presentan especialización horizontal. Por otro lado, el sistema se desea que sea descentralizado y que los roles tengan flexibilidad y capacidad de decisión sobre las tareas que van a realizar, luego se obtiene una ampliación vertical.

Posteriormente se analizan las **restricciones** existentes sobre el comportamiento de los miembros de la organización y los mecanismos de coordinación y cooperación. La dimensión organizativa **coordinación** establece que los mecanismos de coordinación a tener en cuenta son: adaptación mutua, supervisión directa y normalización [Wagner y Hollenbeck, 2004].

La adaptación mutua en los MAS se aplica a través de protocolos de negociación. Las técnicas de negociación se clasifican en [Nwana y Jennings, 1996]: negociación basada en la teoría del juego, que consiste en un proceso interactivo de ofertas y contra-ofertas en las que cada agente escoge un acuerdo que maximiza el valor de utilidad esperado, negociación basada en planes, en la que los agentes planifican sus actividades por separado y después se coordinan empleando un agente de coordinación y negociación inspirada en los humanos; y técnicas de aproximación de inteligencia artificial.

En la supervisión directa, un miembro asume la responsabilidad del trabajo, adquiriendo la autoridad para decidir las tareas a realizar, quién las realizará y cómo se relacionarán para dar lugar al resultado final.

Por último la normalización coordina el trabajo y proporciona estándares y procedimientos que ayudan a determinar cómo realizar las tareas, minimizando la necesidad de comunicación entre las entidades.

Los mecanismos de coordinación utilizados son la adaptación mutua, debido a la ampliación vertical y a la utilización del modelo propuesto de coordinación inspirado en organizaciones humanas, estos aspectos implican una negociación que determine el tipo de tarea a realizar según los requisitos del cliente.

La dimensión organizativa **normalización** o formalización indica el grado de estandarización de los puestos de la organización. En las organizaciones humanas existen cuatro mecanismos de normalización: tareas, resultados, habilidades y comportamientos [Wagner y Hollenbeck, 2004].

La normalización de tareas indica la especificación de tareas o procedimientos concretos que los roles deben realizar para cumplir con sus responsabilidades.

La normalización de resultados refleja formalmente los objetivos o metas esperadas. La normalización de habilidades implica la especificación de cualidades, conocimientos y habilidades necesarias para desarrollar las tareas.

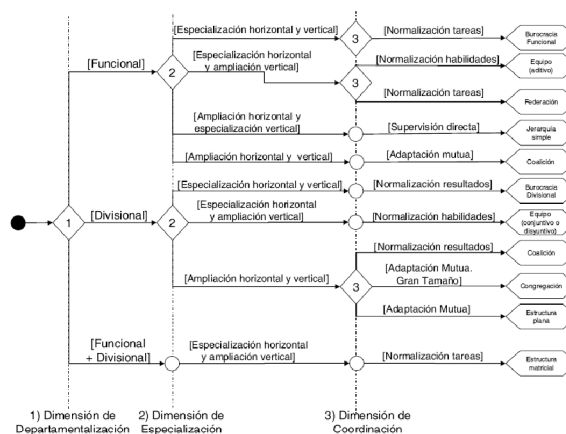
Por último, la normalización de comportamientos consiste en compartir un conjunto de creencias que dan lugar a los comportamientos deseados. Será ésta última la más adecuada para nuestro sistema, ya que los agentes de la organización compartirán una base de creencias que luego utilizarán para llevar a cabo sus objetivos. Adicionalmente, se debe controlar el orden en la invocación de los servicios (normalización de las tareas), así como la calidad de los resultados (normalización de los resultados) para ofrecer mayor flexibilidad en las implementaciones de los servicios de los proveedores.

Tabla 11. Documento C. Dimensiones Organizativas del sistema

<i>C. Dimensiones Organizativas</i>
<b>Departamentalización:</b> divisional y funcional. Se agrupa en base a las funciones principalmente (guías, turistas, etc..) pero también a los productos (monumentos, etc.).
<b>Especialización + Centralización:</b> horizontal y vertical. Los roles están muy especializados. El sistema se desea que sea descentralizado y que los roles tengan flexibilidad y capacidad de decisión sobre las tareas que van a realizar, luego se obtiene una ampliación vertical
<b>Coordinación y Formalización:</b> Adaptación mutua. Normalización de tareas y comportamientos..

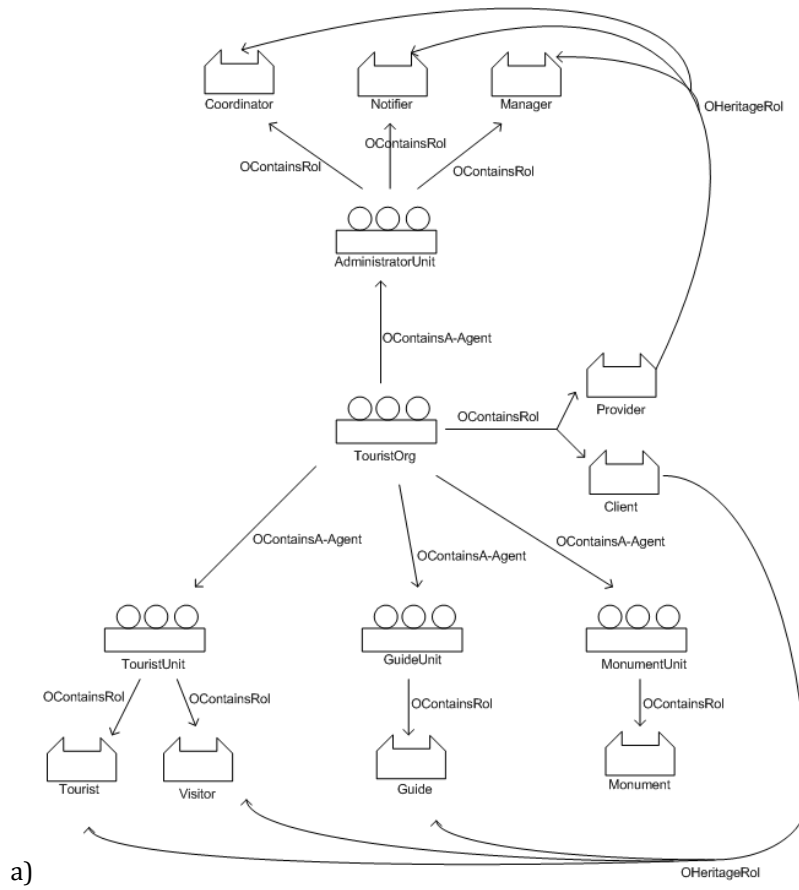
### A.3.2 Fase D. Estructura Organizativa

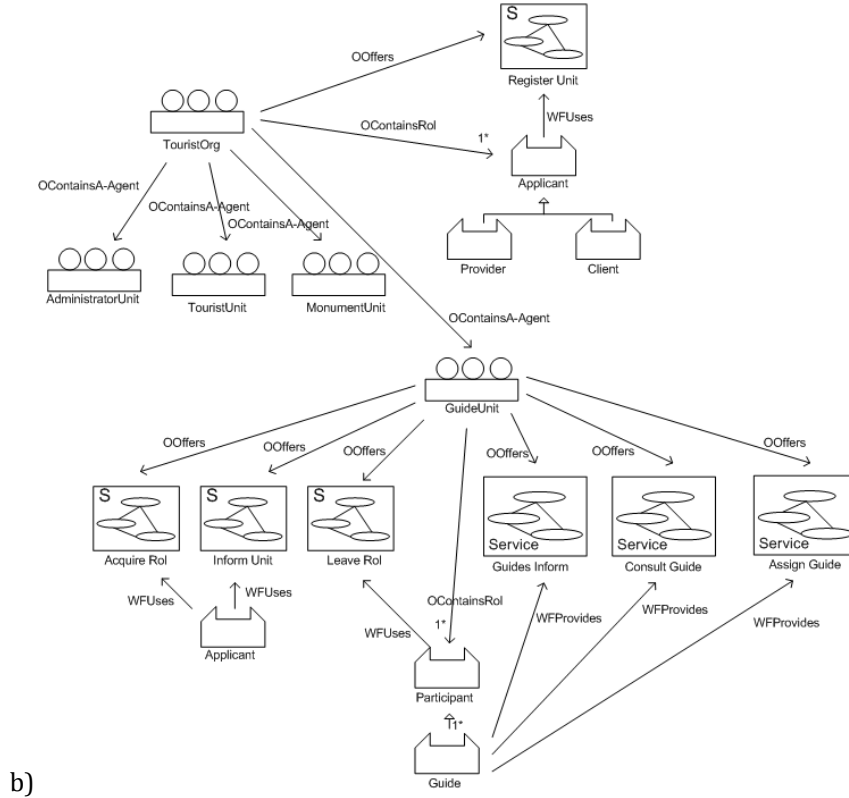
El siguiente paso es la identificación de la estructura que mejor conviene aplicar en el sistema.



**Figura 35. Árbol de decisión de la estructura organizativa**

En este caso, según el árbol de decisión de la fase D [Argente et al., 2008], se ha decidido emplear una estructura de tipo congregación.





**Figura 36. Diagrama actualizado del modelo de organización al adoptar una estructura de tipo congregación. a) Vista estructural, b) Vista funcional (funcionalidad externa).**

En el diagrama de la vista estructural, adaptada conforme al patrón de congregación, la unidad *TouristOrg* contiene tantas unidades como funcionalidades (y productos), en este caso, *ToutistUnit*, *GuideUnit*, *AdministratorUnit* y *MonumentUnit*. Además, los roles proveedor y cliente se refinan dentro de estas nuevas unidades, para especializarse en la funcionalidad o bien en la utilización de los servicios concretos asociados a esos tipos de productos.

En el diagrama de la vista funcional, adaptada también según el patrón de congregación, se muestra solamente el modelado interno de la unidad *GuideUnit* por facilitar su claridad y comprensión. La estructura interna del resto de unidades se ha modelado de forma similar. A través del servicio *Registrar Unidad*, los proveedores pueden solicitar la creación de nuevas unidades organizativas, para así atender la llegada de nuevas funcionalidades o productos.



Las unidades contienen los servicios de *Adquirir Rol*, *Informar Unidad* y *Dejar Rol*, además de los servicios propios dependientes del dominio, que ya habían sido identificados anteriormente. Así por ejemplo, un Guía externo que desee contactar con ellos, debe pasar previamente por un proceso de adquisición de rol correspondiente.

---

## A.4 DISEÑO DE LA DINÁMICA DE LA ORGANIZACIÓN

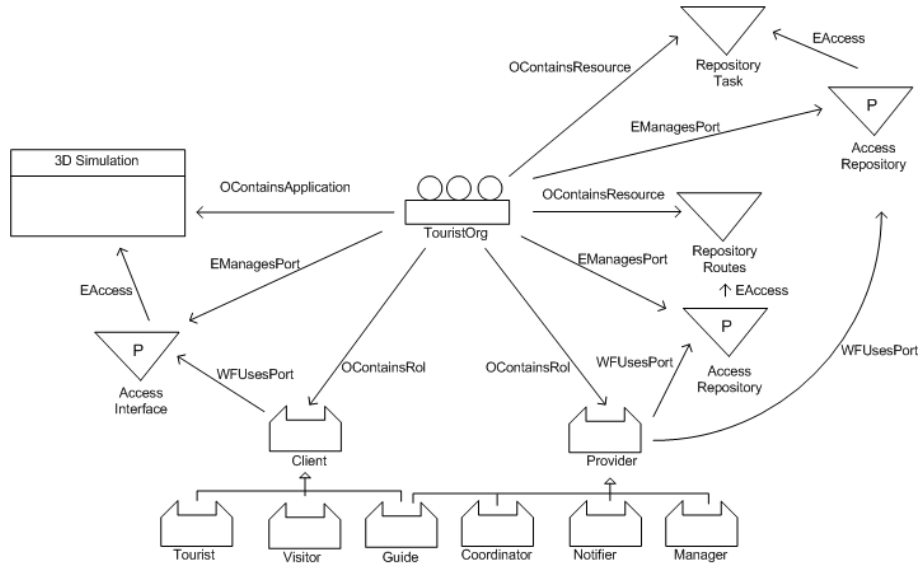
---

### A.4.1 Fase E. Procesos de información-decisión

---

Tras aplicar el patrón de organización seleccionado en la fase anterior, pueden aparecer nuevos servicios asociados a las unidades organizativas, que deben ser descompuestos en tareas. Además, en algunos servicios se indica quién los utiliza pero no quién los proporciona. Por tanto, se revisan los servicios, identificando todos los roles de gestión que se encargan de proporcionarlos.

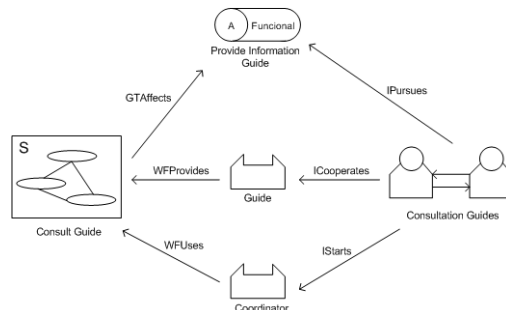
En caso de ser necesario, se crean nuevas entidades recurso para modelar a los repositorios de información (ej. bases de datos), a los dispositivos, a la maquinaria, es decir a cualquier elemento o facilidad requerido para la ejecución de las tareas, sobre los que se apliquen acciones de consumición y producción. Se analizan los procesos de información de la organización, detallando tanto su interacción con el entorno como las interacciones entre las entidades que la componen. En la siguiente figura se muestra el modelo de entorno para la organización *TouristOrg*. Los recursos Repositorio de Tareas y de Rutas se asignan a la unidad *TouristOrg*, pero se crea una entidad *PuertoEntorno* para que el rol proveedor haga uso de los mismos, tanto para lectura como para escritura. Por su parte, para la aplicación de simulación se crea también una entidad *PuertoEntorno*, sobre la que el rol cliente tendrá habilitada la posibilidad de hacer uso de la aplicación, aunque el control de su acceso está regido por la unidad *TouristOrg*.



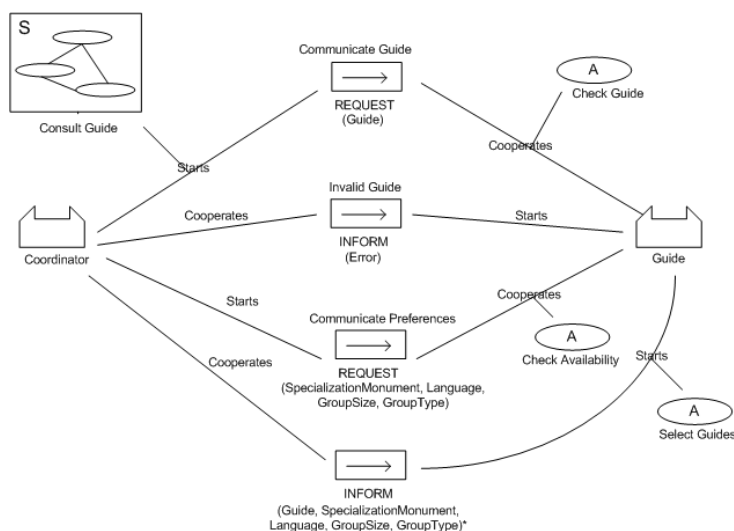
**Figura 37. Diagrama actualizado del modelo de entorno de la organización TouristOrg.**

Respecto a las interfaces con los usuarios, es aquí donde se definen de modo más concreto las aplicaciones de actuación sobre el entorno. Es aquí donde se hace un estudio exhaustivo de las posibilidades de simulación tridimensional para agentes en funcionamiento.

Las interacciones producidas por la activación de los servicios se detallan en el modelo de interacción, donde se asocian los objetivos correspondientes del servicio con la interacción. Se genera también una primera especificación de la interacción, con diagramas de colaboración, donde se indican los mensajes a enviar. En la Figura 38 se muestra el diagrama del *modelo de interacción* para el servicio "Consultar Guía". En dicho diagrama se indican los roles que participan en la interacción y la ejecución del servicio que provoca su activación. En la Figura 39 se muestra el diagrama de colaboración asociado a esta interacción, donde se detalla el modelo de proceso propuesto, indicando los mensajes que se envían las entidades participantes.



**Figura 38. Diagrama del modelo de interacción para el servicio "Consultar Guía"**



**Figura 39. Diagrama de colaboración de la interacción Consultar Guía**

También es en esta fase donde se define la ontología propia del dominio sobre el que se trabaja. Hasta nuestros días se han desarrollado un gran número de ontologías para el turismo y el e-turismo. En nuestro caso de estudio se ha hecho uso de las ontologías *travel.owl* desarrollada en *Protégé*<sup>42</sup>. En particular, en nuestro estudio se ha combinado *travel.owl* con la ontología OnTour [Prantner, 2004] creada por el grupo de investigación DERI (*Digital Enterprise Research Institute*).

Como actividad final de esta fase, se revisan las relaciones entre las distintas unidades organizativas, para así crear nuevos flujos de información entre ellas (denominados vínculos laterales) que faciliten sus relaciones y el intercambio de la información (especialmente sobre el entorno), de modo que se eviten los cuellos de botella y se gestionen mejor posibles fallos de funcionamiento de los agentes.

Una vez se ha realizado el análisis de los procesos de información y decisión, se procede a determinar la "apertura" del sistema, es decir, qué roles son accesibles por agentes externos y qué servicios necesitan ser publicitados para así atraer agentes al sistema.

---

#### A.4.2 Fase F. Dinamicidad del sistema abierto

---

<sup>42</sup> H. Knublauch. <http://protege.stanford.edu/> [Último acceso 20/12/2009]

En esta fase se establece la funcionalidad ofrecida como sistema abierto, que incluye tanto los servicios que se deben publicitar como las políticas de adquisición y liberación de roles. Se determina qué funcionalidad debe ser implementada por agentes internos al sistema y cuál se publicita para poder ser suministrada por agentes externos. De este modo, se modela la dinamicidad de la organización como sistema abierto.

En nuestro caso, los roles de gestión y los servicios asociados a las unidades organizativas estarán disponibles gracias a la plataforma THOMAS. El OMS ofrece los servicios necesarios para el adecuado funcionamiento de una organización. Dichos servicios se clasifican en: servicios estructurales, que modifican la especificación estructural y normativa propia de la organización; y servicios dinámicos, que permiten la gestión de la entrada y salida dinámica de agentes dentro de la organización, así como la adopción de roles. El OMS facilita una serie de servicios para registrar o desregistrar componentes estructurales, en concreto, los roles, las normas y unidades existentes en el sistema. También ofrece servicios para informar sobre cuáles son dichos componentes. Los servicios dinámicos gestionan la creación de agentes nuevos en la organización, la entrada y salida dinámica de los agentes en las unidades del sistema, así como la adopción de roles.

Así, se actualiza el diagrama del modelo de entorno, incluyéndolo dentro del *framework* de ejecución de la plataforma THOMAS. Allí podrán generarse las unidades identificadas para el sistema, los agentes podrán llevar a cabo y publicitar sus servicios y será posible la entrada y salida de nuevas entidades gracias a las funcionalidades proporcionadas en THOMAS.

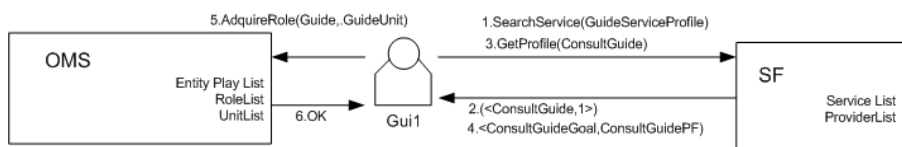
Se determina qué servicios, de los requeridos por la organización, podrán ser proporcionados no sólo por agentes propios del sistema (implementados por los desarrolladores del mismo), sino también por agentes externos.

En este caso, se requiere publicitar los servicios *Generar Ruta, Información Monumento e Información Guías*, cuyo perfil se ha descrito lo suficientemente general para que todos aquellos agentes que trabajen en el dominio de los viajes sean atraídos al sistema. En este caso está claro que será fácil acceder a la organización agentes externos que quieran llevar a cabo un rol de guía o quieran ser un nuevo monumento. Para reflejar la publicitación de los servicios, THOMAS utiliza servicios internos que llevarán a cabo los pasos necesarios para que esos agentes formen parte del sistema con el rol apropiado (informando y quedando registrado mediante el OMS) y para que publiciten sus servicios (en el SF).

La dinamicidad del sistema en THOMAS puede quedar reflejada en diferentes escenarios: registro de nuevos agentes, de nuevos servicios, peticiones de servicios, expulsiones del sistema, etc.

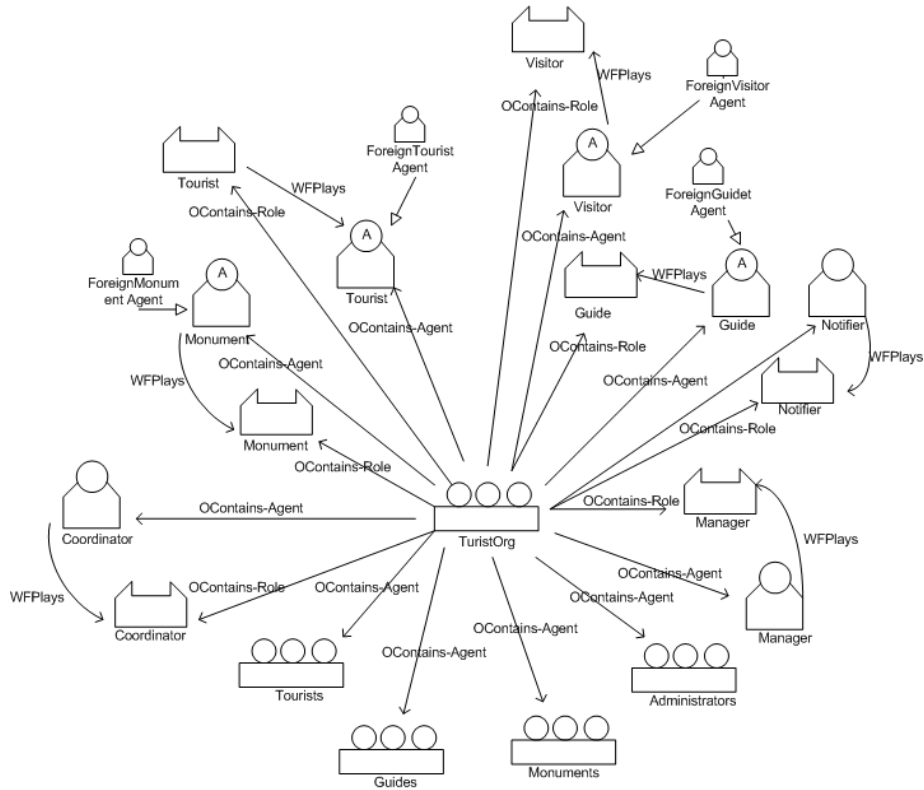
En la siguiente figura se muestra el escenario en que un nuevo Guía es registrado en la organización. Gui1 se ha registrado como miembro de la plataforma THOMAS y pregunta al SF qué servicios ya definidos tienen un perfil similar al suyo. Esta petición es llevada a cabo usando el servicio *SearchService* (mensaje 1) del SF, donde *GuideServiceProfile* corresponde al perfil del servicio de búsqueda implementado por el guía Gui1.

El SF devuelve los identificadores de servicios que satisfacen los requerimientos de búsqueda junto con un valor (*ranking value*) por cada servicio (mensaje 2). *Ranking value* indica el grado de adecuación entre el servicio y el servicio especificado propuesto. Entonces, Gui1 ejecuta *GetProfile* (mensaje 3) para obtener información detallada sobre el servicio *ConsultGuide*. Las salidas del servicio son "service goal" and "profile" (mensaje 4). El perfil del *ConsultGuide* especifica que ese servicio debe ser proporcionado por agentes que juegue el rol *Guide* dentro de *GuideUnit*. Por este motivo, Gui1 solicita al OMS el servicio *AcquireRole* para adquirir ese rol *Guide* (mensaje 5). El servicio *AcquireRole* es llevado a cabo satisfactoriamente (mensaje 6), ya que *GuideUnit* es accesible desde la organización virtual; finalmente Gui1 es registrado como Guía.



**Figura 40. Ejemplo de adquisición de rol por parte de agente externo en THOMAS**

Los roles Monumento, Turista, Visitante y Guía son accesibles por agentes externos, requiriéndose un proceso de adquisición del rol. Los roles Manager, Notificador y Coordinador se asocian a agentes internos, siendo no accesibles. En la Figura 41 se muestra el modelo de organización, indicando los agentes externos (A-Agentes), los internos (Agentes) y los roles que juegan.



**Figura 41. Diagrama actualizado del modelo de organización, en el que se muestran los agentes internos, externos y los roles que juegan.**

Las políticas de adquisición de roles, en la unidad TouristOrg se llevarán a cabo siguiendo las pautas de THOMAS, como podemos ver en el ejemplo de la Figura 40. Así, si por ejemplo un nuevo guía desea registrarse dentro de esa unidad, se le informa de los perfiles y procesos de los servicios que puede utilizar y de las normas que le afectan.

Respecto a las normas a definir, se implementarán internamente como parte de las acciones o tareas de los agentes. Se obliga a los proveedores de servicios a registrar una funcionalidad asociada a cada uno de los servicios que, como mínimo, deben ofrecer. En nuestro caso, el rol guía tiene la obligación de proporcionar, como mínimo, el servicio de *ConsultGuide* correspondiente. Como ejemplo, se indica a continuación las normas asociadas a la funcionalidad mínima que los guías deben ofrecer, así como el control de la consistencia de los servicios que se registran. Para la descripción de las normas, se ha utilizado la notación del lenguaje normativo propuesto en [Argente et al., 2008], que es el lenguaje empleado en la plataforma THOMAS.

---



---

```
OBLIGED Guide REGISTER ConsultGuide
  PROFILE ConsultGuide PROCESS ?p BEFORE deadline
```

**NormaFuncionalidadMínima**

```
FORBIDDEN NuevoMonumento REGISTER CulturalInformation
  PROFILE ?ProfileM PROCESS ?TaskM
  IF NOT( InformationMonumento  $\subseteq$  ?ProfileM AND
  CulturalInformationProcess  $\subseteq$  ?TaskM)
```

**NormaConsistenciaServicioMonnumento**

---

*A.4.3 Fase G. Sistema de medición, evaluación y control*

---

En esta fase se procede a establecer las normas relativas al control de la secuencia de las invocaciones de los servicios, así como a la calidad esperada de sus resultados. Por tanto, se establecen normas para controlar que no se realicen rutas turísticas si antes no se ha efectuado una consulta de guías, o que no se resuelva una incidencia si antes no se ha llevado a cabo su informe asociado.

```
OBLIGED ?a:Coordinator REQUEST GenerateRoute
  BEFORE ?a REQUEST ConsultGuide
```

```
OBLIGED ?a:Manager REQUEST SolveIncidence
  BEFORE ?a REQUEST IncidenceInform
```

**NormasControl**

Respecto a la normalización de resultados, la calidad del servicio debería atender al enfoque de “calidad como satisfacción de las expectativas del cliente”. Para ello sería necesario añadir una nueva funcionalidad al sistema (ej. Recomendar Guía, Recomendar Ruta), a través de la cual los turistas y visitantes pudieran indicar en qué grado las rutas sugeridas por el sistema han satisfecho sus necesidades. Esta información serviría después, por un lado, para que el coordinador mejorara en su proceso de selección de guías y en la generación de las rutas. Por otro, permitiría a la plataforma THOMAS establecer un mecanismo para expulsar a aquellos guías cuyos productos no hayan sido de agrado para un número determinado de turistas. Por simplicidad, en este caso de estudio no entraremos a efectuar ningún tipo de normalización de resultados.

---

*A.4.4 Fase H. Sistemas de recompensas*

---

La funcionalidad interna de los servicios será responsabilidad de los agentes que los ofrezcan, aunque es el sistema el que especifica el perfil de los servicios proporcionados, así como normas sobre el orden de solicitud de servicios, o

bien sobre los resultados a ofrecer. De este modo, ante comportamientos ilícitos o incorrectos de los proveedores (o de los clientes), el sistema actuará en base a sanciones. Se puede analizar el comportamiento global que se desea potenciar entre los agentes del sistema. Se puede hacer uso de recompensas individuales o de recompensas del sistema.

Con la recompensa de sistema se ofrecen gratificaciones generales por el hecho de pertenecer a la organización. En este caso, se podrían ofrecer ofertas y reducción de precios a los turistas, no ya por las rutas realizadas, sino por la cantidad de servicios que utilicen (lo que sería un indicativo del tiempo que permanecen en el sistema de forma activa). En nuestro caso de estudio no se implementa este tipo de sistema de recompensa.

Con la recompensa individual se fomenta principalmente el comportamiento adecuado de los individuos, e incluso que trabajen sobre unos niveles mínimos. Por tanto, resultaría adecuada emplearla sobre los agentes guía, recompensando en base a la cantidad de rutas que realicen y al grado de satisfacción que los turistas y visitantes muestren por ellas. El modo de recompensarles en nuestro caso de estudio es almacenando el agente Manager el grado de satisfacción de los turistas mediante los servicios identificados "Recomendar Ruta" y "Recomendar Guía". Otra forma podría ser mediante gratificación monetaria o bien estableciendo grupos dinámicos (tipo coalición), en los que estuvieran los mejores guías y con los que se generarían las rutas principalmente.



# Anexo B HERRAMIENTAS UTILIZADAS

## B.1 EMFGORMAS

Como guía metodológica se ha utilizado GORMAS. El proceso de análisis y diseño de la organización del caso de estudio puede verse en el Anexo A de este documento. Para este proceso se ha utilizado la herramienta EMFGormas.

EMFGormas<sup>43</sup> es un nuevo enfoque para el modelado de SMA abiertos orientados a servicios que utiliza la tecnología de Eclipse MDA (*Model Driven Architecture*)<sup>44</sup> [Soley, 2005]. Ofrece una herramienta CASE basada en un meta-modelo unificado para la ingeniería de sistemas abiertos en los que las entidades constituyentes interactúan entre ellas por medio de servicios.

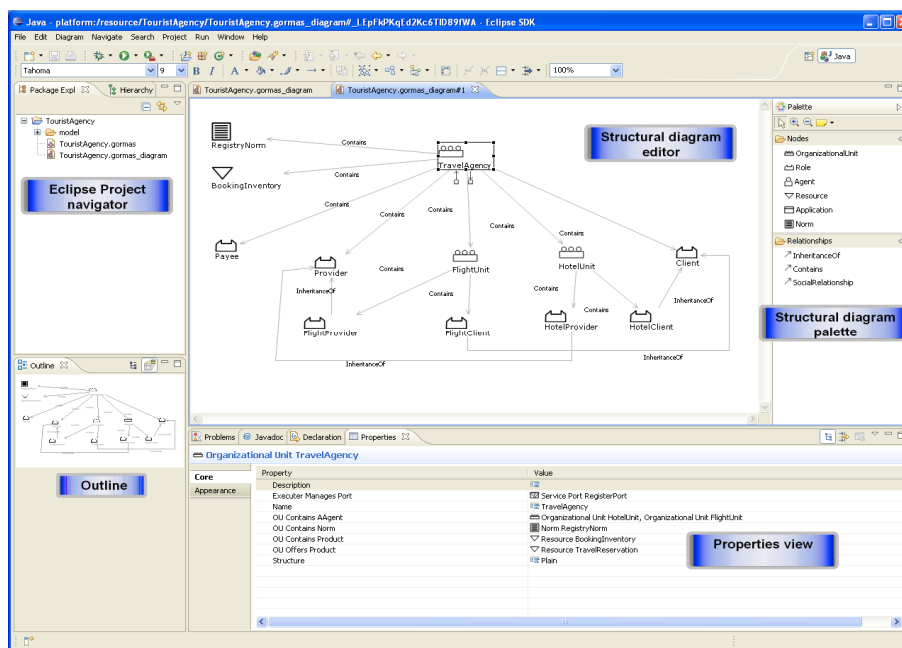


Figura 42. Interfaz EMFGormas e Eclipse

<sup>43</sup> <http://users.dsic.upv.es/grupos/ia/sma/tools/EMFGormas/index.html> [Último acceso 28/1/2010]

<sup>44</sup> <http://www.eclipse.org/modeling/> [Último acceso 28/1/2010]

La iniciativa MDA propone un enfoque de desarrollo de software basado en el modelado y en el mapeo automático de los modelos de origen a modelos destino. Los modelos que representan un sistema y su entorno puede ser visto como los modelos origen y el código fuente puede ser visto como un modelo de destino. Siguiendo los estándares de MDA, la plataforma Eclipse es una iniciativa de código abierto que ofrece un entorno de trabajo para crear este tipo de herramientas. La propia plataforma Eclipse está organizada como un conjunto de subsistemas (implementados en uno o más plugins). La siguiente figura muestra algunos de los plugins como EMF, GEF, GMF, Mofscript, Xpand2 que son ofrecidos por la plataforma para permitir la creación de herramientas basadas en la especificación de metamodelos.

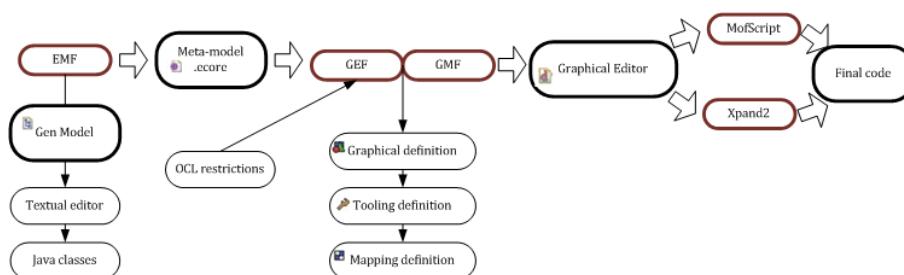


Figura 43. Estructura de plugins de Eclipse

## B.2 THOMAS

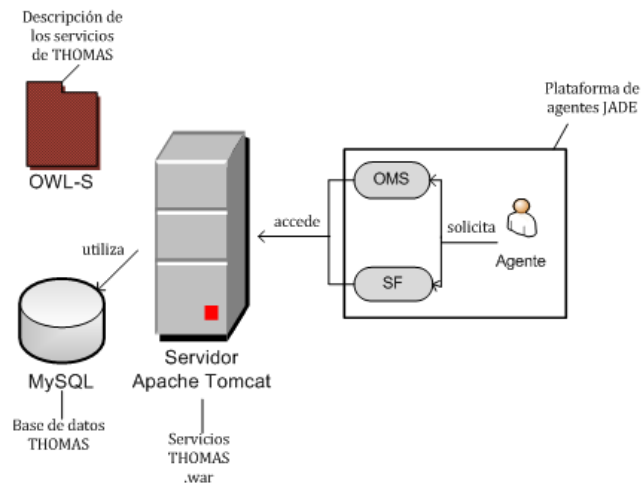
Aunque la descripción de la arquitectura ya se ha hizo en el apartado 3.1.2.1, este punto pretende ahondar en los componentes técnicos de la misma.

La aplicación THOMAS<sup>45</sup> está formada por dos componentes fundamentales:

- ❖ Los Servicios Web, que implementan los servicios de la plataforma. Estos servicios están descritos en el documento de la arquitectura abstracta.
- ❖ Los agentes intermediarios (SF, OMS), que actúan de traductores entre los agentes FIPA de la plataforma y los Servicios Web. Este tipo de agentes reciben peticiones de servicios mediante el protocolo FIPA-request y se encargan de acceder a los Servicios Web correspondientes. Esta solicitud de los servicios Web se realiza empleando el API de mindswap.

La siguiente figura muestra un esquema de la aplicación THOMAS.

<sup>45</sup> <http://www.dsic.upv.es/users/ia/sma/tools/Thomas/index.html> [Último acceso 2/2/2010]



**Figura 44. Aplicación THOMAS**

Los **servicios** proporcionados hacen uso de una base de datos MySQL<sup>46</sup> que es necesario configurar previamente. Tanto en el ejemplo ejecutable como en el código suministrado asume que la base de datos se encuentra en la misma máquina en la que se proporcionan los servicios (localhost). En el caso de no ser así es necesario modificar los archivos *.java* de los servicios donde se configura el acceso a la base de datos.

La implementación de las **normas** implica que al registrar una nueva norma dentro de la plataforma se lleven a cabo las acciones necesarias para poder controlarla. De modo que en la implementación del servicio *RegisterNorm* cada vez que se solicita el registro de una nueva norma se analiza la norma y se llama al módulo de gestión de normas (*NormativeManager*) que se encargará de llevar a cabo las acciones necesarias en función del tipo de norma. El primer paso necesario para considerar la gestión de normas es desarrollar un analizador que nos permita determinar si una norma es válida con respecto a la especificación BNF de la gramática normativa. Este proceso de análisis Léxico-Sintáctico es llevado a cabo por la clase *NormativeLanguageParser*. Esta clase ha sido generada de forma automática por el generador de analizadores sintácticos *JavaCC (Java Compiler Compiler)* a partir de la gramática del lenguaje normativo. *JavaCC* genera un *parser* para una gramática presentada en notación BNF en código Java. La salida proporcionada por la función principal (llamada *Norm*) de la clase *NormativeLanguageParser* es el árbol de parseado correspondiente a una frase dada. Dicho árbol contiene una serie de anotaciones que serán empleadas por el módulo analizador para determinar el tipo de norma.

<sup>46</sup> <http://www.mysql.com/> [Último acceso 10/01/2010]

De este modo, el módulo de análisis (*NormativeChecker*) se encarga de *parsear* una norma, haciendo uso del módulo anteriormente comentado, y analizar el árbol de análisis sintáctico para determinar si se trata de una norma de tipo conocido. La función principal de esta clase (*analyzeNorm*) devuelve un array que contiene el tipo de la norma (*incompatibility*, *maxCardinality*, o *Unknown*) y la información contenida en la misma.

Ejemplo: Normas Incompatibilidad

```
FORBIDDEN role1ID REQUEST AcquireRole  
MESSAGE(CONTENT(role 'role2ID'))47
```

La plataforma muestra una interfaz gráfica en la que es posible gestionar los componentes de la misma (SF;OMS, etc):

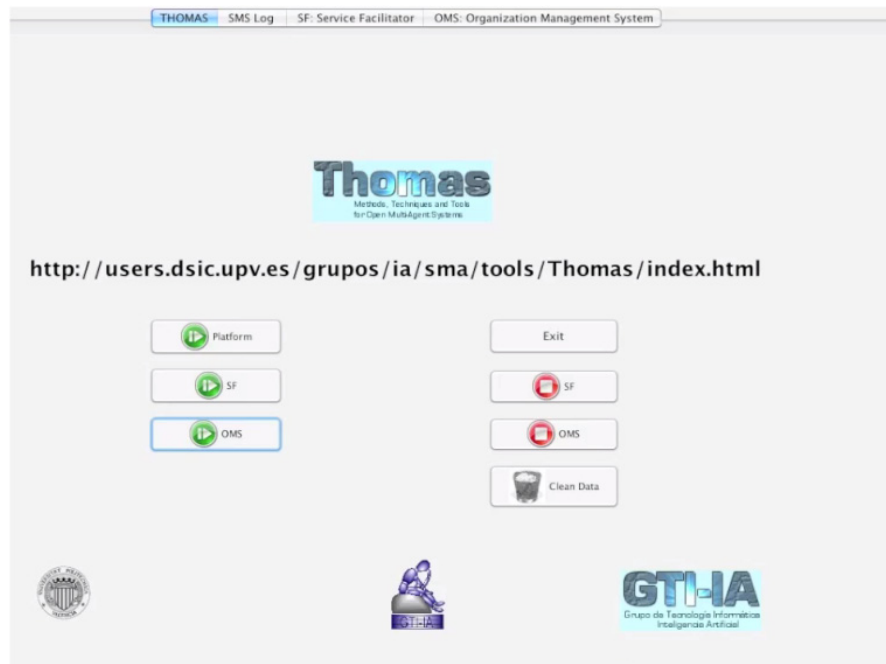


Figura 45. Interfaz THOMAS

---

<sup>47</sup> Para solicitar el registro de una norma es necesario sustituir los espacios en blanco entre las distintas palabras por “\_”.

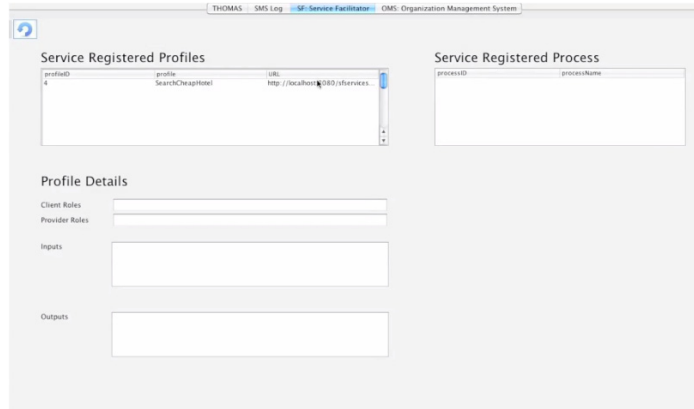


Figura 46. Interfaz manejo de SF en THOMAS

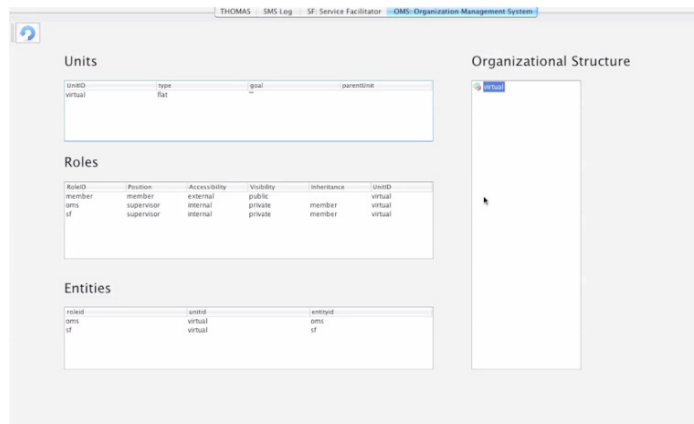


Figura 47. Interfaz manejo de OMS en THOMAS

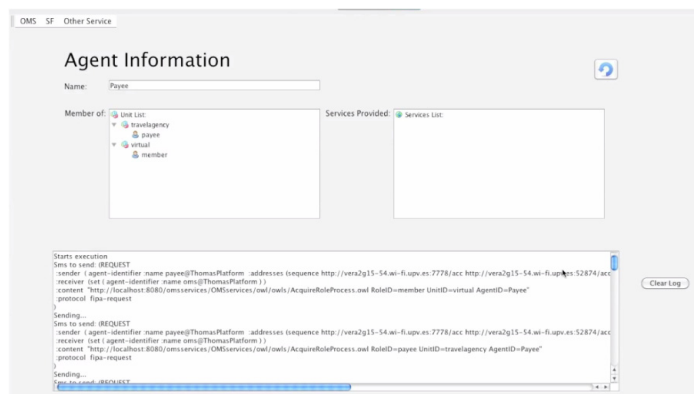


Figura 48. Interfaz manejo de agentes e interacciones en THOMAS

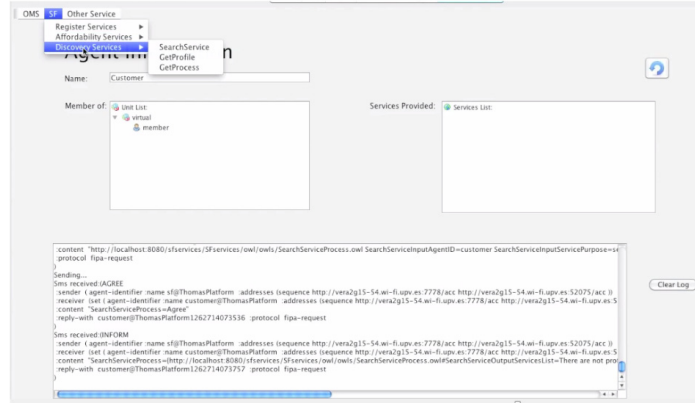


Figura 49. Interfaz manejo de servicios en THOMAS

### B.3 REPAST

Dentro de las tecnologías complementarias estudiadas en este trabajo de tesis se incluye una rama de investigación nueva centrada en la simulación y representación de agentes de manera virtual. Actualmente están apareciendo herramientas y mecanismos para poder llevar a cabo este tipo de actuaciones.

Para la simulación virtual del caso de estudio del capítulo 6 se estudiaron algunas de las herramientas más actuales que permiten llevar a cabo simulaciones en tres dimensiones. La siguiente tabla muestra algunas de ellas junto con sus propiedades principales.

Tabla 12. Comparación herramientas de simulación

Propiedad/Herramienta	JAS3D <sup>48</sup>	REPAST <sup>49</sup>	MASON <sup>50</sup>	breve <sup>51</sup>
Desarrolladores	Jacksonville State University	University of Chicago, USA	George Mason University, USA	University of Goteborg, Sweden
Fecha creación	2006	2000	2003	2006
Sistema Operativo	Windows, UNIX, Linux, Mac OSX	Windows, UNIX, Linux, Mac OSX	Windows, UNIX, Linux, Mac OSX	Windows, UNIX, Linux, Mac OSX
Simulación 2D	No proporciona simulación 2D	Proporciona simulación en 2D	Proporciona simulación en 2D	Proporciona simulación en 2D
Simulación 3D	Proporciona simulación en 3D	Proporciona simulación en 3D	Proporciona simulación en 3D	Proporciona simulación en 3D
Simulación con	si	si	si	si

<sup>48</sup> Java Agent Simulator 3D - <http://mcis.jsu.edu/faculty/agarrett/jas3d/index.html> [Último acceso 2/2/2010]

<sup>49</sup> REPAST Agent Simulation Toolkit - <http://REPAST.sourceforge.net/> [Último acceso 2/2/2010]

<sup>50</sup> Multi-Agent Simulator Of Neighborhoods or Network <http://www.cs.gmu.edu/~eclab/projects/mason/> [Último acceso 2/2/2010]

<sup>51</sup> Breve: <http://www.spiderland.org/> [Último acceso 2/2/2010]

<b>un agente</b>				
<b>Simulación con varios agentes</b>	Si. Añadiendo agentes mediante la interacción con el usuario y en tiempo de ejecución.	Si. Añadiendo agentes mediante la interacción con el usuario y en tiempo de ejecución o mediante alguna función Java en el código fuente.	Si. Pero no se puede añadir agentes en tiempo de ejecución.	Si. Pero no se puede añadir agentes en tiempo de ejecución.
<b>Cambio de comportamiento en los agentes</b>	El cambio es proporcionado a través de una clase "comportamiento" donde se puede modificar y añadir nuevas acciones en tiempo de ejecución y también mediante la interacción con el usuario.	El cambio de comportamiento puede ser definido a través del código fuente, antes de que el programa sea compilado. No se puede cambiar el comportamiento mediante interacción con el usuario y en tiempo de ejecución, como en la Jas3D	No proporciona cambio de comportamiento en tiempo de ejecución.	No proporciona cambio de comportamiento en tiempo de ejecución. Solamente a través del código fuente antes de la compilación del programa.
<b>Interacción con usuario</b>	Si. Se puede añadir agentes y comportamiento s mediante interacción con el usuario.	Sí, Se puede añadir agentes y definir sus enlaces dentro de una determinada red.	No proporciona interacción con usuario. Son ejemplos de simulación predefinidas y solo se puede sacar y poner los agentes de la visualización y no del entorno de simulación.	Los agentes solo interaccionan con otros agentes y con el entorno.
<b>Creación de entorno (carga de mapas, etc...)</b>	Si. Pero solamente se puede modificar la estructura del entorno como, tamaño (anchura y altura), obstáculos, y cuadrados	Si. Se puede crear mapas con diferentes entornos y con dimensiones distintas.	Sí	Sí
<b>Importación de modelos 3D</b>	No se pueden importar modelos 3D.	Se puede importar modelos del tipo: XML, Obj, ms3d, .Jws, lwo	No se pueden importar modelos 3D.	No se pueden importar modelos 3D.
<b>Diseño de agentes (mapas conceptuales)</b>	No.	Proporciona una herramienta para la representación de propiedades, agentes, etc. gráficamente.	No.	No.
<b>Agentes interaccionan con obstáculos y sensores</b>	Si- mediante diez sensores siendo uno sonar y uno infrarrojo.	Los agentes interaccionan con el entorno, a través da la posición en el Grid. Ellos no poseen sensores.	Si, los agentes pueden interaccionar con los obstáculos y sensores.	Si, los agentes interaccionan con el entorno y con otros agentes a través de sensores.
<b>Agentes</b>	No. Los agentes	Los agentes	Si. Pero no hay	Si. Pero no hay cambio

## Anexos

<b>interaccionan con otros agentes</b>	reconocen otros agentes como un obstáculo, los sensores en ello solo detectan los colores blanco, azul, negro y verde.	interaccionan a través de un red particular, pero no hay un cambio de mensajes entre ellos.	cambio de mensajes entre los agentes.	de mensajes entre los agentes.
<b>Utilización con JADE</b>	No	No	No	No
<b>Facilidad de programación</b>	Si., Para añadir el comportamiento.	No es tan sencilla pero los diagramas para diseño de los agentes y de su comportamiento ayudan en la programación. Además existen modelos que se puede utilizar en código fuente.	No.	Relativamente sencilla. La herramienta trae muchos ejemplos y una documentación que ayuda bastante al programador.
<b>Documentación de la herramienta</b>	Completa y de fácil entendimiento	Completa y de fácil entendimiento	Incompleta.	Completa y de fácil entendimiento
<b>Tipos de agente que simula</b>	Agentes simples con algunos sensores.	Agentes hacia bioinformática, simulación geográfica, partículas y otros	Agentes hacia bioinformática, simulación geográfica, partículas y otros.	Agentes hacia bioinformática, simulación geográfica, partículas y otros.
<b>Lenguajes de programación utilizados</b>	Java	Java	Java	Python o Steve
<b>Entorno para el desarrollo de ejemplos</b>	Cualquier herramienta de programación con soporte al lenguaje Java.	Eclipse	Cualquier herramienta de programación con soporte al lenguaje Java.	El entorno para desarrollo ya está insertado en la herramienta Breve.
<b>Integración con otras herramientas(estadística, matemática, minería de datos y etc.)</b>	No.	Integración con: Matlab, VisAD, ORA, R statistics environment, Pajek, JoSQL, Weka, iReport, JUNG, Microsoft Excel y UCINET	Integración con Quicktime a través de la herramienta Java Media Studio.	No.

La herramienta que proporciona más ventajas es *REPAST Symphony*, pues posee una variedad de herramientas integradas en ella. REPAST es una herramienta de código libre. Inicialmente fue implementado exclusivamente en Java, pero en las últimas versiones tiene tres núcleos: REPAST J (REPAST para Java); REPAST.Net (REPAST para Microsoft.Net); y REPAST Py (REPAST para Python). Sin embargo, los tres núcleos han sido sustituidos por lo que se conoce hoy en día como REPAST Symphony (REPAST S). Las principales ventajas que ofrece REPAST son las características de modelado de agentes de manera gráfica (ver Figura 50), y la mejora de la interfaz de usuario (ver Figura 51).



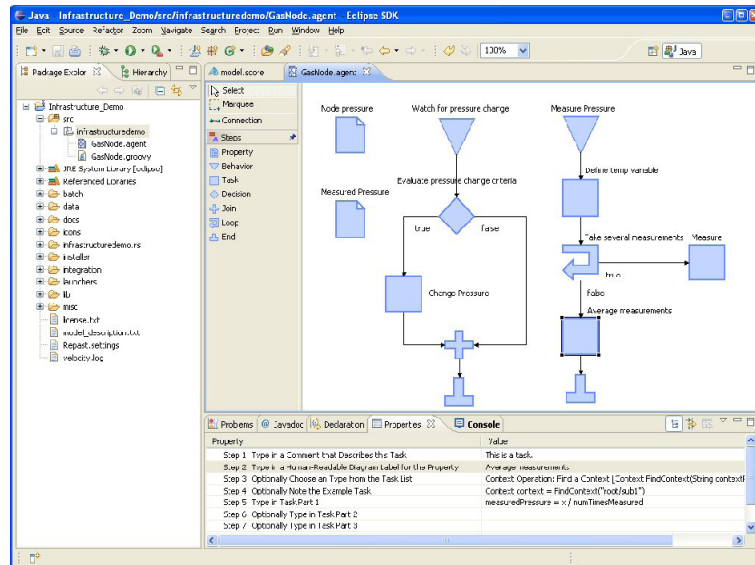


Figura 50. Desarrollo de modelos en REPAST S

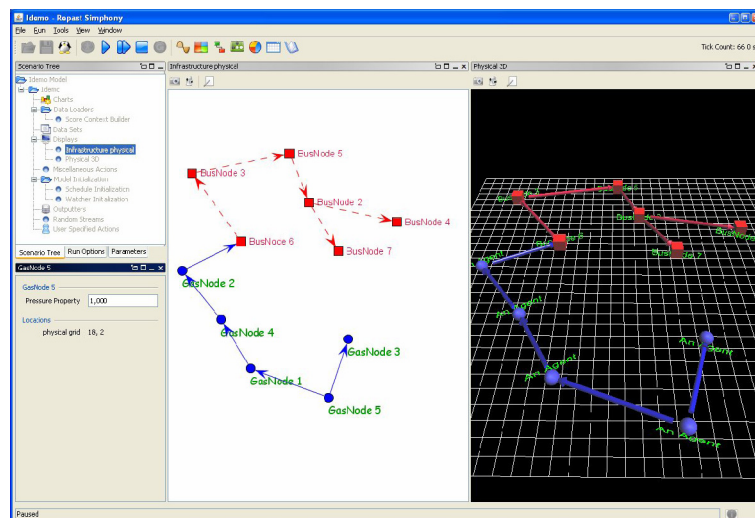


Figura 51. Interfaz de usuario en REPAST S

Las desventajas, de todas las herramientas estudiadas, es que no se pueden integrar con la plataforma JADE, pues no están conforme a los estándares de la *Foundation for Intelligent Physical Agents*(FIPA).

Para ello, en este trabajo se propone combinar la plataforma de agentes JADE con REPAST proporcionando así un entorno para la simulación de modelos de agentes. JADE, siguiendo los estándares de FIPA, es utilizada para la creación y

desarrollo de SMA. REPAST fue creada como una plataforma de propósito general para la simulación de cualquier tipo de agente de manera gráfica. Es importante no confundir "los sistemas de agentes" que pueden ser simulados en REPAST, con los sistemas multi-agente creados en JADE.

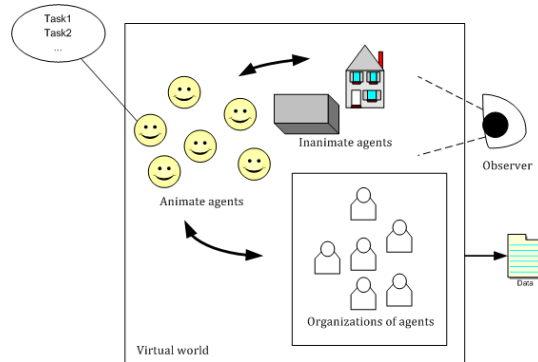


Figura 52. Simulación

La combinación de ambas tecnologías hace posible la simulación de organizaciones de agentes cuya ejecución puede ser observada por los usuarios y cuyos resultados de la simulación pueden ser analizados y estudiados. Para la combinación de las dos tecnologías principalmente se llevaron a cabo las siguientes acciones: (i) creación de una clase especial de agente; (ii) creación de un comportamiento de sincronización especializado; (iii) modificación del esquema de los mensaje ACL de JADE para incluir aspectos temporales y de simulación; (iv) implementación de una clase "MiddleBehaviour" (comportamiento) especial en Jade para usar las funciones de visualización proporcionadas por REPAST; (v) capacidad de llevar una contabilización del tiempo de simulación.

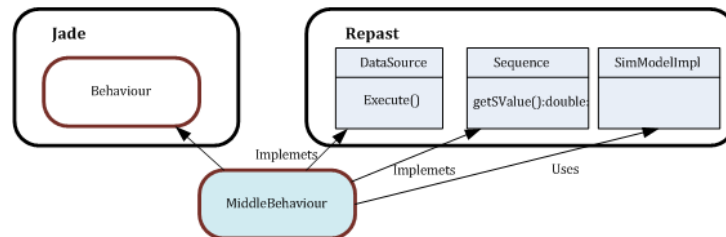


Figura 53. Combinación JADE-REPAST.

Un sistema construido en REPAST está formado por un conjunto de objetos, principalmente los agentes descritos por el modelo, y además, el propio modelo de simulación que se encarga de inicializar y controlar toda la simulación. En la práctica se comporta como un simulador de eventos discretos cuya unidad de tiempo se llama *tick*.

La implementación de un modelo basado en agentes en REPAST se hace en las siguientes etapas:

1. La programación de todas las clases-agente (variables-estados y métodos-comportamientos) del sistema.
2. La programación del conjunto de métodos computacionales que serán utilizados para representar gráficamente aquellas variables del sistema que el modelador considere oportunas (estadísticos o datos individuales de los agentes), así como las salidas de datos en ficheros planos uqe pueden ser utilizados para un posterior análisis estadístico o en ficheros de simulación gráfica en formato video o imagen.
3. La programación de secuencia (*scheduling*) de eventos que se suceden durante un paso de la simulación.

Estas tres etapas coinciden con los métodos principales de la clase `SimModelImpl` responsable de la ejecución y control de la simulación. La describe el diagrama de las principales clases de nuestro modelo implementado en Repast. `SimModelImpl` es responsable de ejecutar y gobernar cada simulación. `GuideAgent` y `TouristAgent` implementan la simulación de los agentes Guías y Turistas de la organización. En la simulación, cada Guía estará asociado a uno o más turista (un grupo) para realizar las rutas.

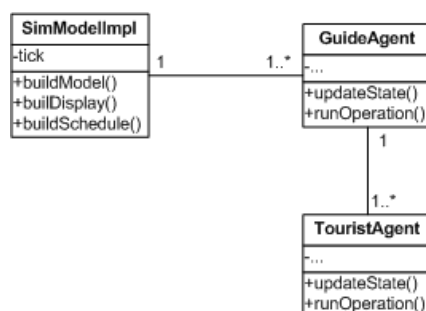


Figura 54. Diagrama de las principales clases del modelo de simulación implementado en Repast.

## B.4 JADE

JADE<sup>52</sup> (Java Agent Development Framework) es la plataforma de agentes que permite el desarrollo de sistemas multiagente, permite “vivir” y “convivir” a los agentes dentro de ella y además, la comunicación entre ellos.

<sup>52</sup> <http://jade.tilab.com> [Último acceso 10/01/2010]

Sus características principales son:

- ❖ Está totalmente implementada en JAVA. Esto facilita la portabilidad y la movilidad de los agentes entre distintas plataformas, independientemente de la plataforma del sitio de destino.
- ❖ Es software libre distribuido por TILAB en código fuente bajo LPGL.
- ❖ Cumple con las especificaciones FIPA
  - Arquitectura: Esto nos dará muchas ventajas a la hora de la integración de diferentes aplicaciones, incluso con plataformas de diferentes propietarios.
  - Lenguaje de comunicación empleado FIPA-ACL.
  - Servicios de agentes: ciclo de vida, páginas blancas, páginas amarillas, transporte de mensajes, etc.
  - Conjunto de herramientas gráficas que soportan la depuración y ejecución de agentes (RMA, sniffer, etc.).

Su arquitectura básica se muestra en el siguiente esquema.

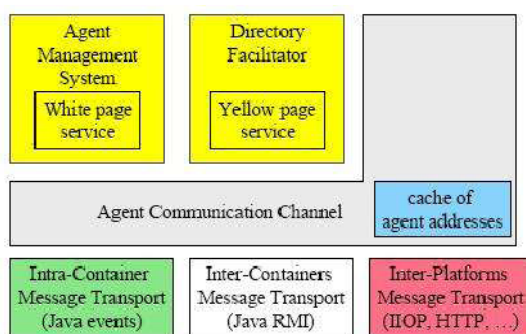


Figura 55. Arquitectura JADE

AMS . Agent Management System:

- ❖ Garantiza que cada agente en la plataforma tenga un único nombre.
- ❖ Encargado de proporcionar los servicios de páginas blancas y ciclo de vida, y de mantener el directorio de los identificadores de agentes (AID: Agent Identifier) y su estado.
- ❖ Cada agente debe registrarse con el AMS para obtener un AID válido.

DF . Directory Facilitator:

- ❖ Agente que proporciona el servicio de páginas amarillas.
- ❖ Un agente puede encontrar otros agentes que proporcionan los servicios que requiere para cumplir sus objetivos.

ACC . Agent Communication Channel:

- ❖ Software que controla el intercambio de mensajes.

---

## B.5 HERRAMIENTAS DE MODELADO

---

Para los modelos 3D utilizados esREPAST, se utilizaron herramientas adicionales de diseño y modelado en tres dimensiones. Las más características son:

- ❖ Google SketchUp<sup>53</sup>: herramienta para crear modelos en 3D desarrollada por Google.
- ❖ Maya 2009<sup>54</sup>: software de Autodesk para el modelado, la animación, la creación de efectos visuales y la renderización en 3D.

Ambas herramientas permiten modificar, crear y exportar modelos tridimensionales en distintos formatos como por ejemplo *.obj*, que pueden posteriormente ser importados desde REPAST (gracias a la utilización del API Java 3D).

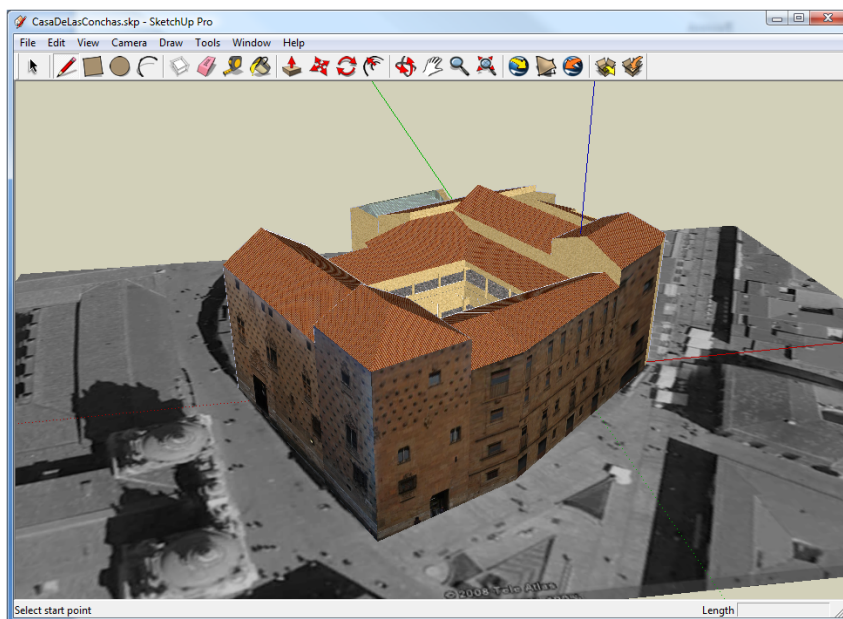


Figura 56. Google SketchUp

---

<sup>53</sup> <http://sketchup.google.com/intl/es/> [Último acceso 10/01/2010]

<sup>54</sup> <http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=13577897> [Último acceso 10/01/2010]

## ANEXO C. PROYECTOS, PUBLICACIONES Y TRABAJOS RELACIONADOS

Los dos proyectos de investigación relacionados directamente con el desarrollo de esta tesis son:

- ❖ Proyecto THOMAS<sup>55</sup> (“THOMAS: Métodos Técnicas y Herramientas para Sistemas Multiagente Abiertos”) TIN2006-14630-C03-03, otorgado por el Ministerio de Ciencia e Innovación.

Los factores sociales en las organizaciones de sistemas multi-agentes, como hemos visto a lo largo de este trabajo, son cada vez más importantes para estructurar las interacciones en mundos abiertos y dinámicos. En esta interesante área, enmarcada en la más general de Sistemas Inteligentes, podemos identificar dos grandes líneas de trabajo:

- Dinamicidad/Regulación: flexibilidad para permitir la entrada y salida de los agentes, evolución de la estructura organizativa, mecanismos de regulación, etc
- Heterogeneidad: diferentes tipos de agentes con capacidades diversas, coordinación en tiempo de ejecución (requiere descripción semánticas de las capacidades/servicios), diferentes dispositivos (recursos físicos de los dispositivos), diferentes canales de comunicación (wireless, wireline, etc)

El objetivo de este proyecto es avanzar y aportar soluciones en estas líneas, principalmente en los aspectos relacionados con estructuras organizativas.

- ❖ Proyecto OVAMAH<sup>56</sup>: Organizaciones Virtuales Adaptativas: Mecanismos, Arquitecturas y Herramientas. TIN 2009-13839-C03-03, otorgado por el Ministerio de Ciencia e Innovación. Proyectos de Investigación Fundamental No Orientada.

El principal objetivo de este proyecto es el estudio y propuesta de nuevos métodos, herramientas y/o mecanismos que permitan la evolución y organización de organizaciones virtuales basadas en tecnología de sistemas multi-agente. De tal manera que permita la detección de situaciones no cooperativas o no deseadas mediante la evaluación dinámica del sistema así como su posterior tratamiento

---

<sup>55</sup> <http://www.thomas-tin.org> [Último acceso 9/2/2010]

<sup>56</sup> <http://bisite.usal.es/webisite/?q=es/node/2> [Último acceso 9/2/2010]

para que la organización evolucione de forma autónoma y sea capaz de adaptarse a la nueva situación.

Respecto a la publicaciones, el siguiente listado muestra algunas de las más relevantes relacionadas con la tesis. El listado completo de publicaciones está accesible en: <http://bisite.usal.es>.

Artículos en revistas internacionales:

- ❖ De Paz, J.F., Rodríguez, S., López,V. and Bajo,J. (2010). *An Adaptive Algorithm for Feature Selection in Pattern Recognition*. In Press, ISSN: 0020-7160. International Journal of Computer Mathematics. Taylor & Francis. Ámsterdam (The Netherlands). (JCR-2010) 0.308 .Posición: Acept. (JCR 2010) 170/175; Tercer Tercio (4º Cuartil) (Applied Mathematics).
- ❖ Bajo, J., De Paz, J.F., Rodríguez, S. and González A. (2010). *Multi-Agent System to Monitor Oceanic Environments.*, ISSN: 1069-2509. Integrated Computer Aided Engineering. IOS Press. Columbus (USA) In Press (2010). (JCR-2010) 0.617.Posición: Acept. (JCR 2010) 80/94; Public. (JCR-2010) Tercer Tercio (4º Cuartil) (Comp. Sciences, Artificial Intelligence)(Comp. Sciences, Interdisciplinary Applications) 76/94 Tercer Tercio (4º Cuartil)(Engineering Multidisciplinary) 38/68 Segundo Tercio (3º Cuartil).
- ❖ Rodríguez S., Corchado J.M., Fernandez A., Ossowski S., Julian V., Botti V. (2009) *A THOMAS Based Multi-Agent System for Recommendations and Guidance in Malls*. ISSN: 1888-0258. Journal of Physical Agents. Special Sessions on Practical Applications of Agents and Multiagent Systems. Vol. 3, Núm 2 Pág:21-26. Septiembre 2009.
- ❖ Bajo J., Corchado J.M., De Paz Y., De Paz J.F., Rodríguez S., Martín Q. and Abraham A. (2008). *SHOMAS: Intelligent Guidance and Suggestions in Shopping Centres*. Applied Soft Computing . Elsevier Science. ISSN: 1568-4946, doi:10.1016/j.asoc.2008.11.009. Vol 9, núm, 2, Pág: 851-862. Marzo 2009. Factor Impacto (JRC-2008): 1.909 Puesto 30, Segundo Cuartil en la categoría Ciencia de la Computación e Inteligencia Artificial. Puesto 23, Primer Cuartil en la categoría Computer Science, Interdisciplinary Applications.
- ❖ De Paz J.F., Rodríguez S., Bajo J. and Corchado J.M.(2009). *Mathematical model for dynamic case based planning*. International Journal of Computer Mathematics. Taylor & Francis Ltd. Volumen: 86, num 10,11 Páginas, inicial: 1719 final: 1730 Fecha: Octubre 2009. Factor Impacto (JRC-2008): 0.308 Puesto 170, Cuarto Cuartil en la categoría Mathematics, Applied.
- ❖ Corchado J.M., Bajo J., De Paz J.F. and Rodríguez S. (2009). *An Execution Time Neural-CBR Guidance Assistant*. Neurocomputing. Elsevier Science. Volumen: 72, Num.13-15 Páginas, inicial: 2743 final: 2753 Fecha: Agosto 2009. Factor Impacto (JRC-2008): 1.234 Puesto 53, Tercer Cuartil en la categoría Ciencia de la Computación e Inteligencia Artificial.



Capítulos de libro:

- ❖ Juan F. de Paz, Martí Navarro, Sara Rodríguez, Vicente Julián, Javier Bajo and Juan M. Corchado. (2010). *Temporal Bounded Planner Agent for Dynamic Industrial Environment*. Libro: IEA-AIE 2010. Springer Verlag. Berlin (Germany). June 2010. Top 67 del Computer Science Conference Ranking AI (0.57/1) Posición: 46/701
- ❖ De Paz Juan F., Rodríguez Sara, Sánchez Juan M., de Luis Ana y Corchado Juan M. (2007): *Context Aware Hybrid Agents on Automated Dynamic Environments*. Ref. ISBN: 978-3-540-74971-4 - ISSN:1615-3871,DOI: 10.1007/978-3-540-74972-1\_51 Libro: Innovation in Hybrid Intelligent Systems. Volumen: 44 Páginas, inicial: 25 final: 32. Fecha: Noviembre 2007. Editorial: Advances in Soft Computing Series. Springer Verlag
- ❖ Rodríguez S., Fernández A., Julián V., Corchado J.M., Ossowski S. and Botti V. (2009). *THOMAS-MALL: A Multiagent System for Shopping and Guidance in Malls*. Ref. ISBN: 978-3-540-87655-7 0 revista 1 Libro: Proceedings of IWANN'09. Vol 5517. Pág. 594–601 Fecha: Junio 2009. Editorial: J. Cabestany et al., Springer Verlag. Lecture Notes in Artificial Intelligence (LNAI), Springer Verlag.
- ❖ Bajo J., Tapia D.I., De Luis A., Rodríguez S., De Paz J.F. and Corchado J.M. (2007). *Hybrid Architecture for a Reasoning Planner Agent* Ref. ISSN 0302-9743 Proceedings of KES 2007 Volumen: 4693. Páginas, inicial: 461 final: 468 Septiembre 2007. Editorial: Apolloni et al. (Eds.), Lecture Notes in Artificial Intelligence (LNAI), Springer Verlag.
- ❖ Dante Tapia D.I., Bajo J., Rodríguez S., Manzano J.M. and Corchado J.M. (2007) *Hybrid Agents Based Architecture on Automated Dynamic Environments*. Ref. ISSN 0302-9743 0 revista 1 Libro: Proceedings of KES 2007 Volumen: 4693 Páginas, inicial: 453 final: 460 Fecha: 2007. Editorial: Lecture Notes in Artificial Intelligence (LNAI), Springer Verlag.
- ❖ Bajo J., Corchado J.M. y Rodríguez S. (2007). *Intelligent Guidance and Suggestions using Case-Based Planning*. Ref. ISSN 0302-9743 Libro: Proceedings of ICCBR 2007. Páginas, inicial: 389 final: 403. Agosto 2007. Editorial: Lecture Notes in Artificial Intelligence (LNAI), Springer Verlag..

Artículos en actas de congresos de reconocido prestigio:

- ❖ Martí Navarro, Sara Rodríguez, Vicente Julián, Vivian F. López. (2010). *Open MAS Architecture. Providing Real Time Solutions*. ISBN: 978-3-642-12432-7, Advances in Intelligent and Soft Computing Series, Vol.71, pp. 69-76. Trends in Practical Applications of Agents and MultiAgent Systems: 8th International Conference on Practical Applications of Agents and MultiAgent Systems. Editorial: Yves Demazeau, Frank Dignum, Juan M. Corchado and Javier Bajo (eds). Springer Verlag. Salamanca (Spain). Abril 2010. Factor De Impacto: (JCR-2007) Top 67 del Computer Science Conference Ranking AI (0.56/1) Posición: 50/701



- 
- ❖ Sara Rodríguez, Juan F. De Paz, Pablo Sánchez, Juan M. Corchado. (2010).: *Context-Aware Agents for People Detection and Stereoscopic Analysis*. ISBN: 978-3-642-12432-7, Advances in Intelligent and Soft Computing Series, Vol.71, pp. 173-182. Libro: Trends in Practical Applications of Agents and MultiAgent Systems: 8th International Conference on Practical Applications of Agents and MultiAgent Systems. Editorial: Yves Demazeau, Frank Dignum, Juan M. Corchado and Javier Bajo (eds). Springer Verlag. Salamanca (Spain). Abril 2010. Factor De Impacto: (JCR-2007) Top 67 del Computer Science Conference Ranking AI (0.56/1) Posición: 50/701
  - ❖ Rodríguez S., Pérez-Lancho B., De Paz J.F., Bajo J. and Corchado J.M. (2009). *Ovamah: Multiagent-based Adaptive Virtual Organizations*. Ref. ISBN: 0 revista 1 Libro: 12th International Conference on Information Fusion. Fecha: Julio 2009. Lugar de publicación: Seattle, Washington, USA
  - ❖ Rodríguez S., De Paz J.F., Tapia D.I. and Corchado J.M. (2008) *Using a Distributed Multi-Agent Architecture for Optimizing the Performance of a Case-Based Planning Mechanism*. Ref. ISBN: 978-3-88120-903-8 0 revista 1 Libro: 3rd European Workshop on Case-Based Reasoning and Context-Awareness (CaCoA). ECCBR 2008. Páginas, inicial: 43 final: 52 Fecha: Septiembre 2008. Editorial: Martin Schaaf (ed.). ECCBR Workshops. Lugar de publicación: Berlin (Germany)
  - ❖ Juan F. De Paz, Sara Rodríguez, Juan M. Corchado, Javier Bajo. (2008). *Dynamic Case Based Planning*. Ref. ISBN: 978-84-612-1982-7 Libro: Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering (CMMSE '08) Volumen: 1 Páginas, inicial: 213 final: 223 Fecha: junio 2008. Editorial (si libro): J. Vigo-Aguilar (Eds.)
  - ❖ De Paz J.F., Sánchez J.M., Rodríguez S., González A. and Corchado J.M. (2007). *Guiding Clients in a Shopping Mall using Case-based Planning and FYDPS Neural Networks*. Ref. ISBN: ISBN: 978-84-611-8846-8 0 revista 1 Libro: CAEPIA 2007. Páginas, inicial:181 final:190 Noviembre 2007. Editorial (si libro): Daniel Borrajo, Luis Castillo y Juan Manuel Corchado (Eds.)- Universidad de Salamanca. Lugar de publicación: Salamanca (Spain).

