

Sistemas Multiagente

Tema 4: Sistemas Multiagente



Vicent Botti
Universitat Politècnica de València

GTI IA Grupo de Tecnología Informática
Inteligencia Artificial



Asociación Española para
la
Inteligencia Artificial
(AEPIA)

UIMP Universidad Internacional
Menéndez Pelayo

Tema 4- Índice

4.1 Introducción

¿Qué es un SMA?

¿Cómo funciona un SMA?

Ventajas

Sin estándares no hay agentes

4.2 Modelo Conceptual de Plataforma FIPA

FIPA: Modelo Conceptual de la Comunicación de un Agente

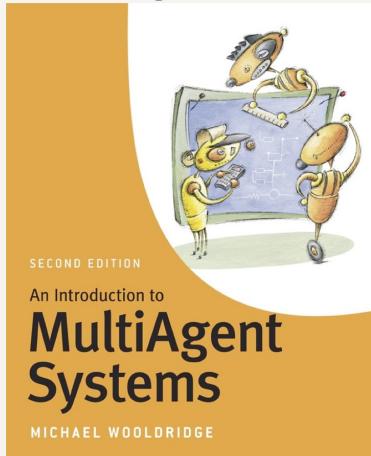
4.3 Plataformas de Agente

4.4 Programando un SMA

4.5 Conclusiones

Bibliografía

Bibliografía básica:



An Introduction to MultiAgent Systems – Second Edition

by Michael Wooldrige

Published May 2009 by John Wiley & Sons

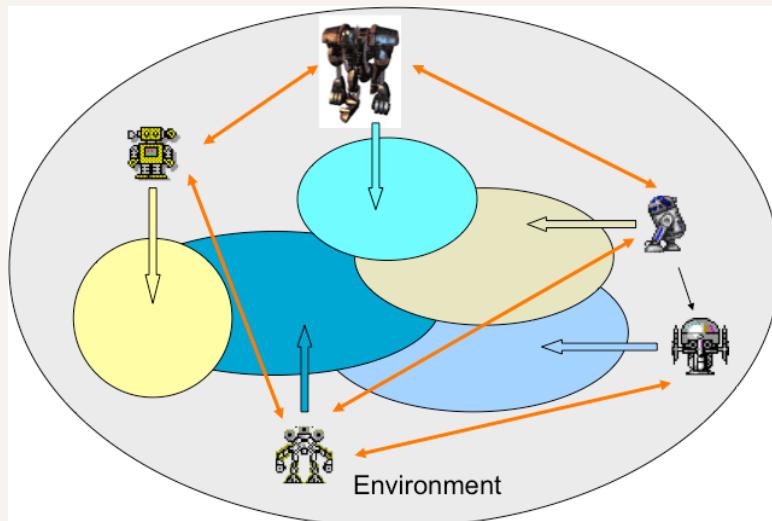
ISBN-10: 0470519460

ISBN-13: 978-0470519462

4.1 Introducción

¡Recordemos!

- Un sistema multiagente está constituido por un conjunto de agentes (dos o más agentes) que interactúan los unos con los otros.
 - Caso más general: los agentes actúan en representación de usuarios que tienen diferentes objetivos y motivaciones.
 - Para interactuar con éxito, requieren capacidades para cooperar, coordinarse y negociar con cada uno de los otros, tal como hace la gente.



Enlaces de interés



<https://www.youtube.com/watch?v=wVy7cSv9rY0>

<https://www.youtube.com/watch?v=TrVmohNaVww>

https://www.youtube.com/watch?v=_NrYlOt5goQ

<https://www.youtube.com/watch?v=uVpN136q7N8&nohtml15=False>

https://www.youtube.com/watch?v=azIOCFjDZWA&feature=youtube_gdata

https://www.youtube.com/watch?v=z40JP-PJ1vI&feature=youtube_gdata

<http://meta-guide.com/videography/100-best-multi-agent-system-videos>

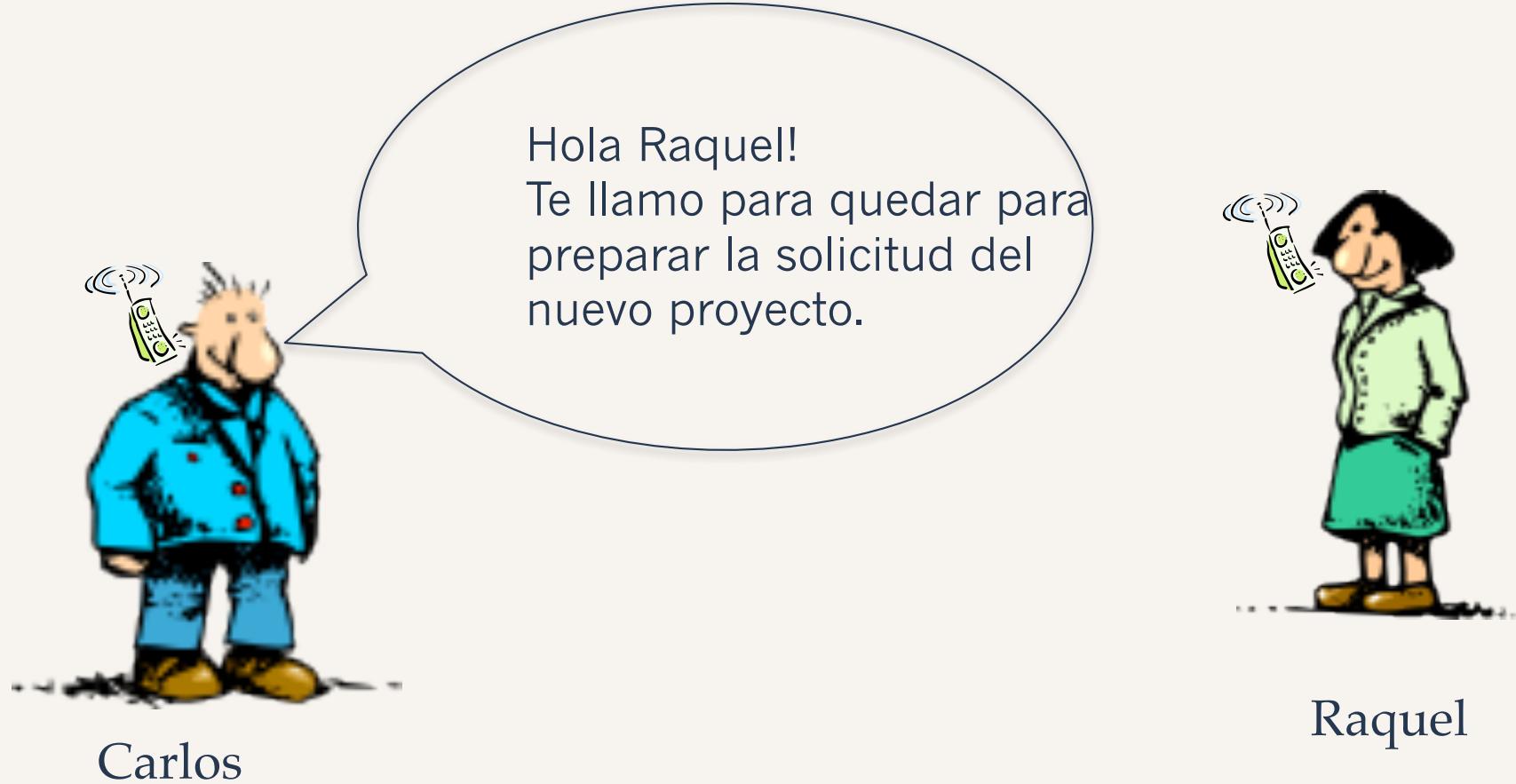
¿Qué es un Sistema Multi-Agente?

Ejemplo:

Una conversación entre Carlos (residente en Valencia) y Raquel (residente en Madrid) para decidir una reunión de trabajo.



¿Cómo funciona un SMA?



¿Cómo funciona un SMA?



Carlos



Carlos escucha y asiente

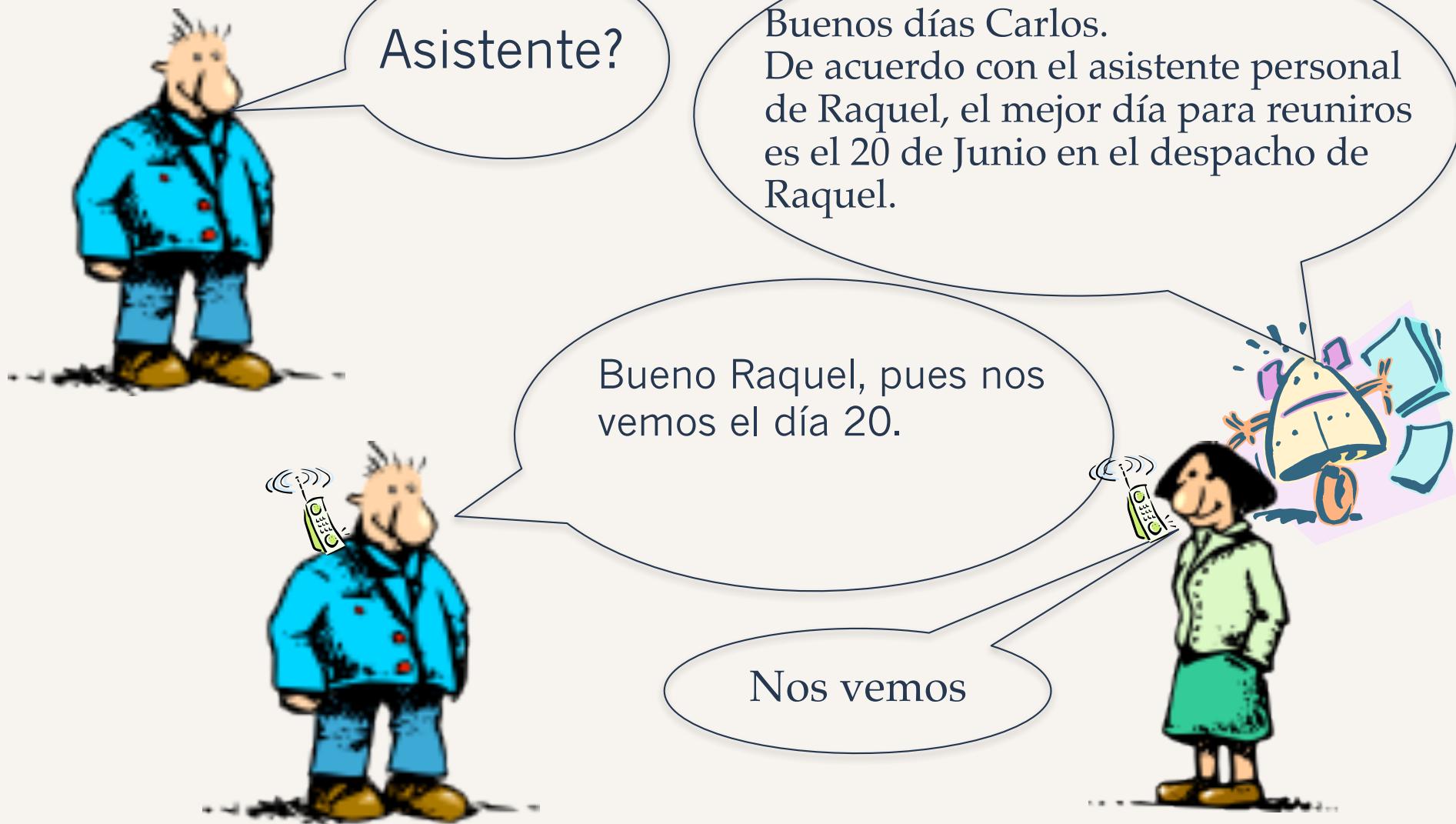
Podemos quedar la
semana que viene,
preguntemos a nuestros
asistentes...



Raquel



¿Cómo funciona un SMA?



¿Cómo funciona un SMA?



Asistente, ¿me organizas el viaje?



Sí Carlos.

Te propongo salir en el AVE a Madrid el 20 de junio a las 7 de la mañana, llegada a Madrid a 18.35, en clase preferente para que puedas desayunar. Vuelta, salida el 21 a las 11 de la mañana, llegada a Valencia a las 12.35. en clase turista
¿De acuerdo?

¿Cómo funciona un SMA?



Sí, perfecto

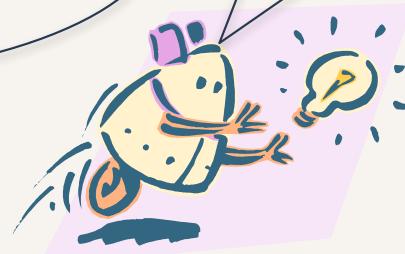


Alojamiento en el hotel Westin Palace,
la noche del 20 al 21
Lo confirmo?

Viaje organizado. Tren y hotel
confirmados.
De la comida y la cena se encarga la
Sra. Raquel.



Sí, muy bien.
Muchas gracias.



¿Cómo funciona un SMA?

- ❖ El agente monitoriza la actividad del usuario
 - ❖ Lee/escucha la conversación del usuario
 - ❖ Reconoce patrones en la conversación
 - ❖ Deduce información y objetivos en función de experiencia pasada

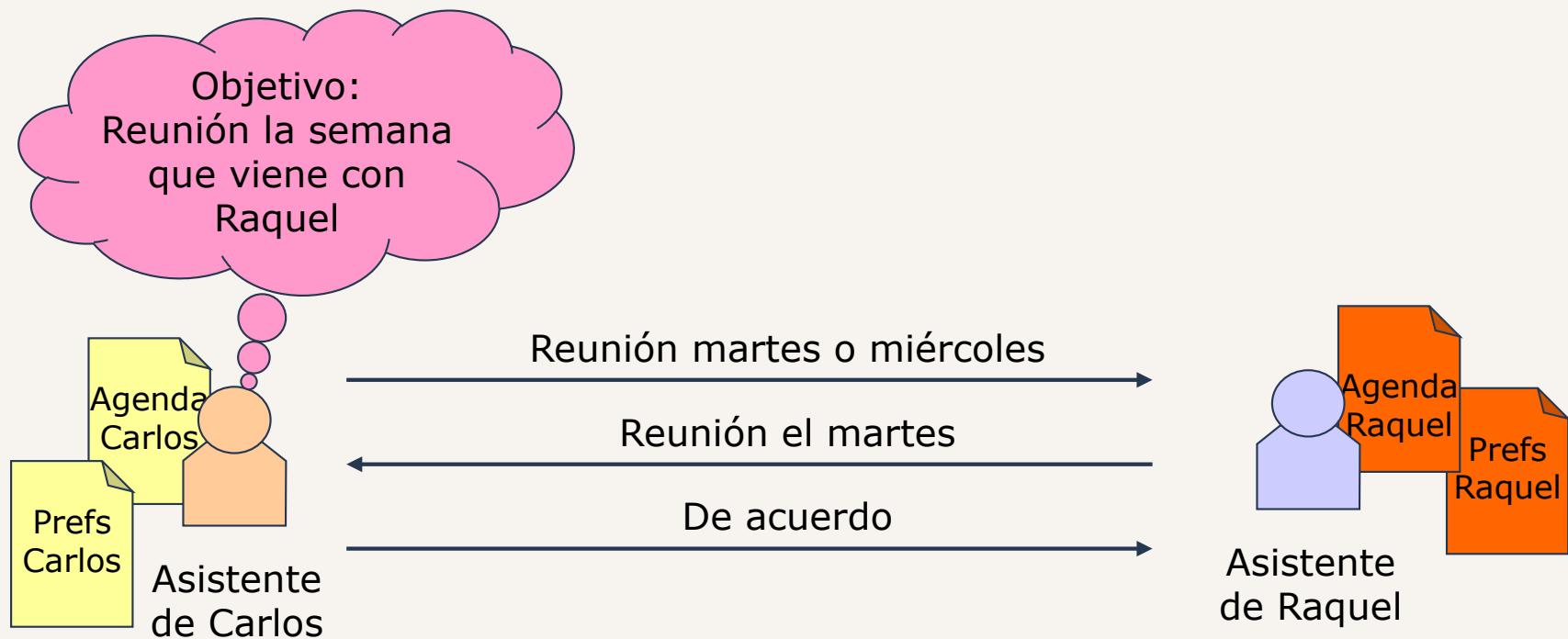


¿Cómo funciona un SMA?

- El agente persigue satisfacer sus objetivos
 - Toma decisiones
 - Puede descomponer objetivos en subobjetivos
 - Ejecuta tareas

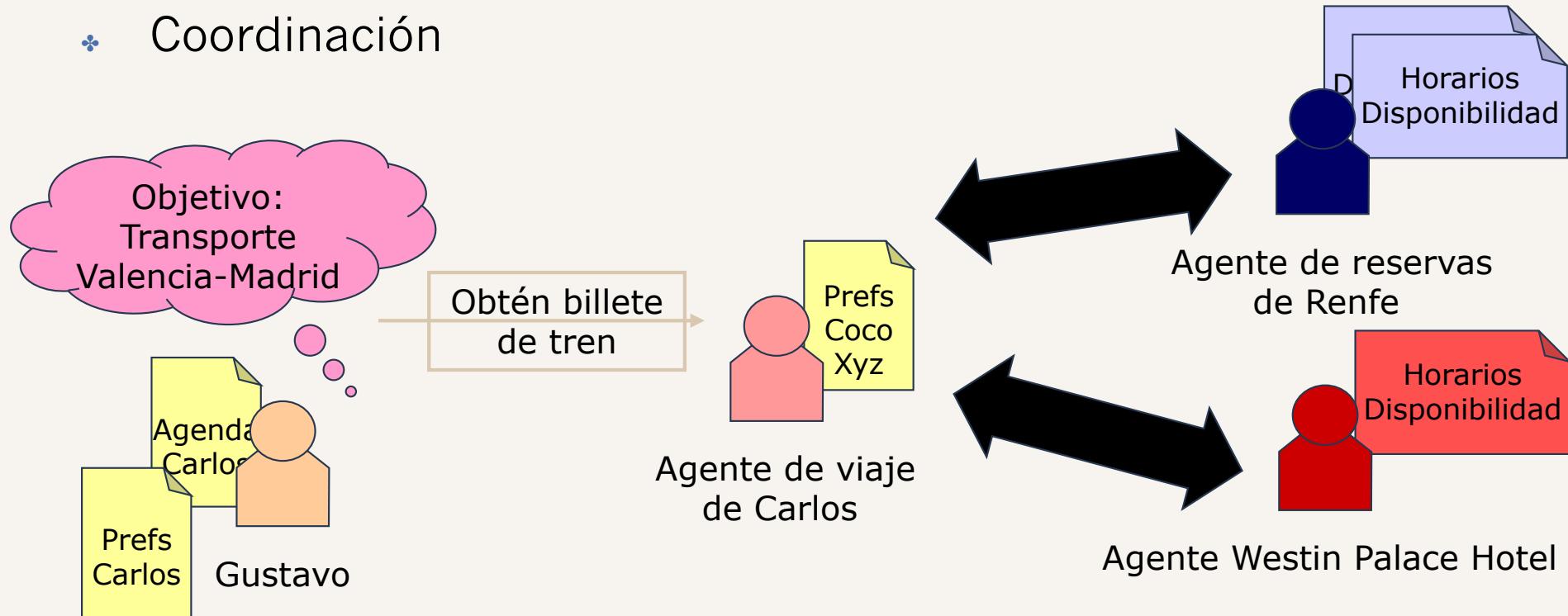
¿Cómo funciona un SMA?

- ✿ Para cumplir objetivos necesita la colaboración con otros agentes
 - ✿ Negociación
 - ✿ Delegación
 - ✿ Coordinación



¿Cómo funciona un SMA?

- Para cumplir objetivos necesita la colaboración con otros agentes
 - Negociación
 - Delegación
 - Coordinación



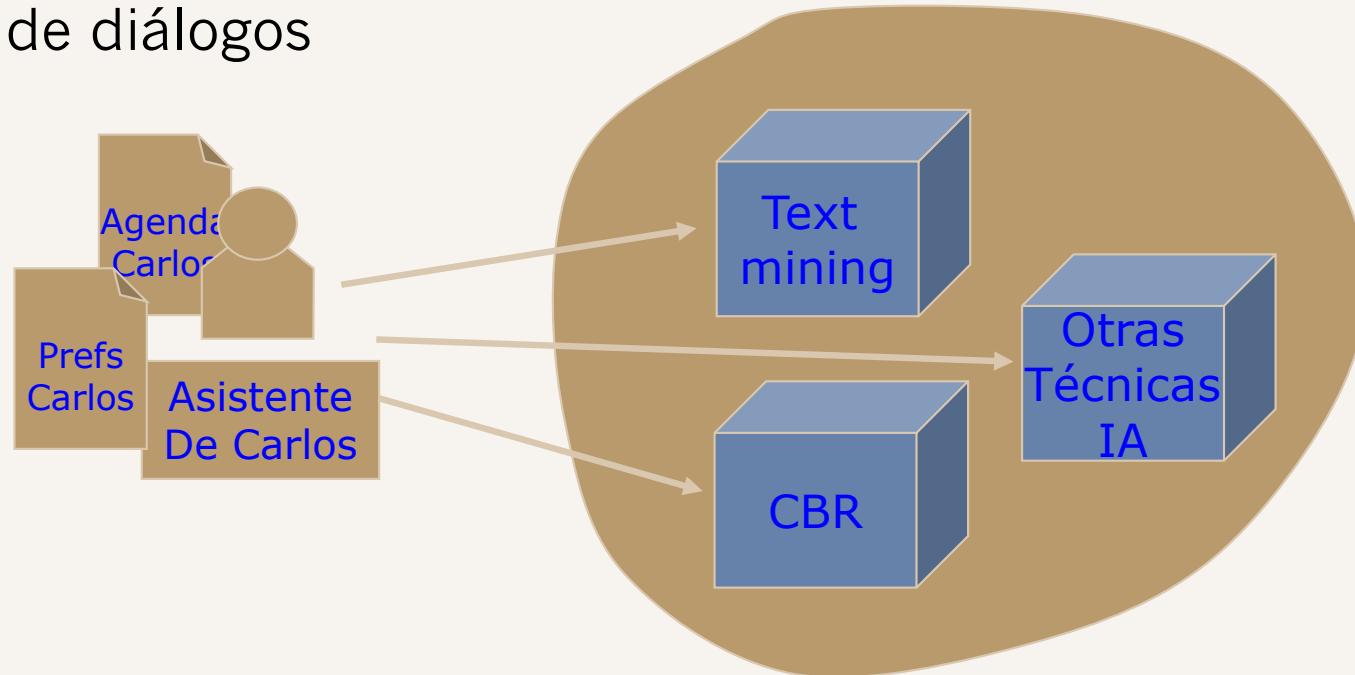
¿Cómo funciona un SMA?

- Los agentes necesitan servicios de localización de agentes
 - Páginas blancas
 - Páginas amarillas



¿Cómo funciona un SMA?

- ❖ Comunicación con el usuario
 - ❖ Interfaces avanzadas
 - ❖ Información implícita a partir de experiencia pasada o preferencias del usuario
 - ❖ Gestión de diálogos



Sistemas Multiagente

Tema 4: Sistemas Multiagente



Vicent Botti
Universitat Politècnica de València

GTI IA Grupo de Tecnología Informática
Inteligencia Artificial



Asociación Española para
la
Inteligencia Artificial
(AEPIA)

UIMP Universidad Internacional
Menéndez Pelayo

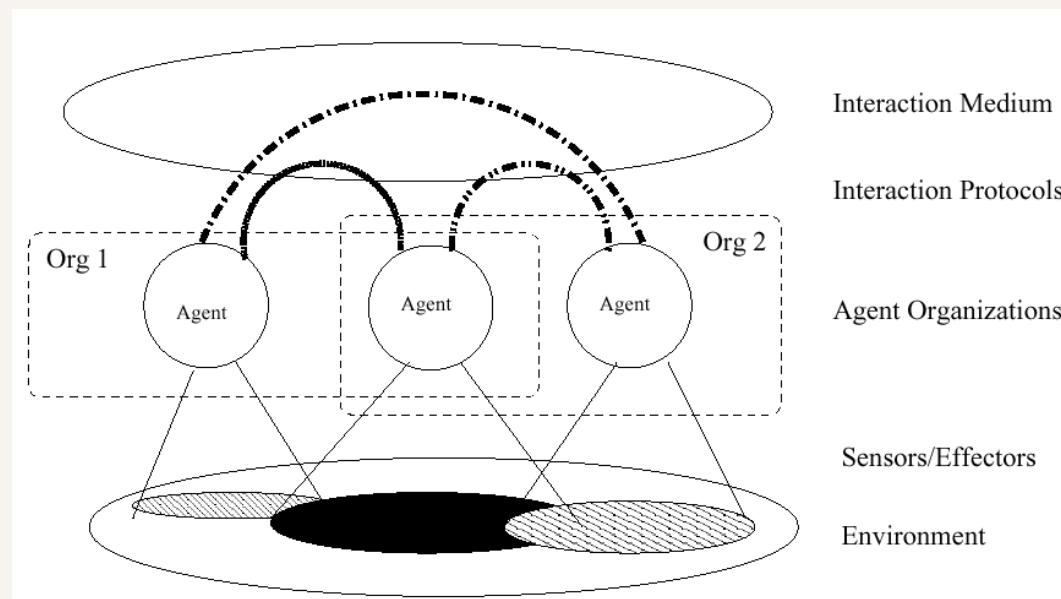
4.1 Introducción



- La tecnología de SMA nos permite:
 - Modelar sistemas reales complejos y con características claramente distribuidas
 - Visión de un sistema como una organización computacional consistente de varios “roles” interactuando.
 - Diseño (Métodos) → Implementación (Lenguajes) → Implantación (Middleware)

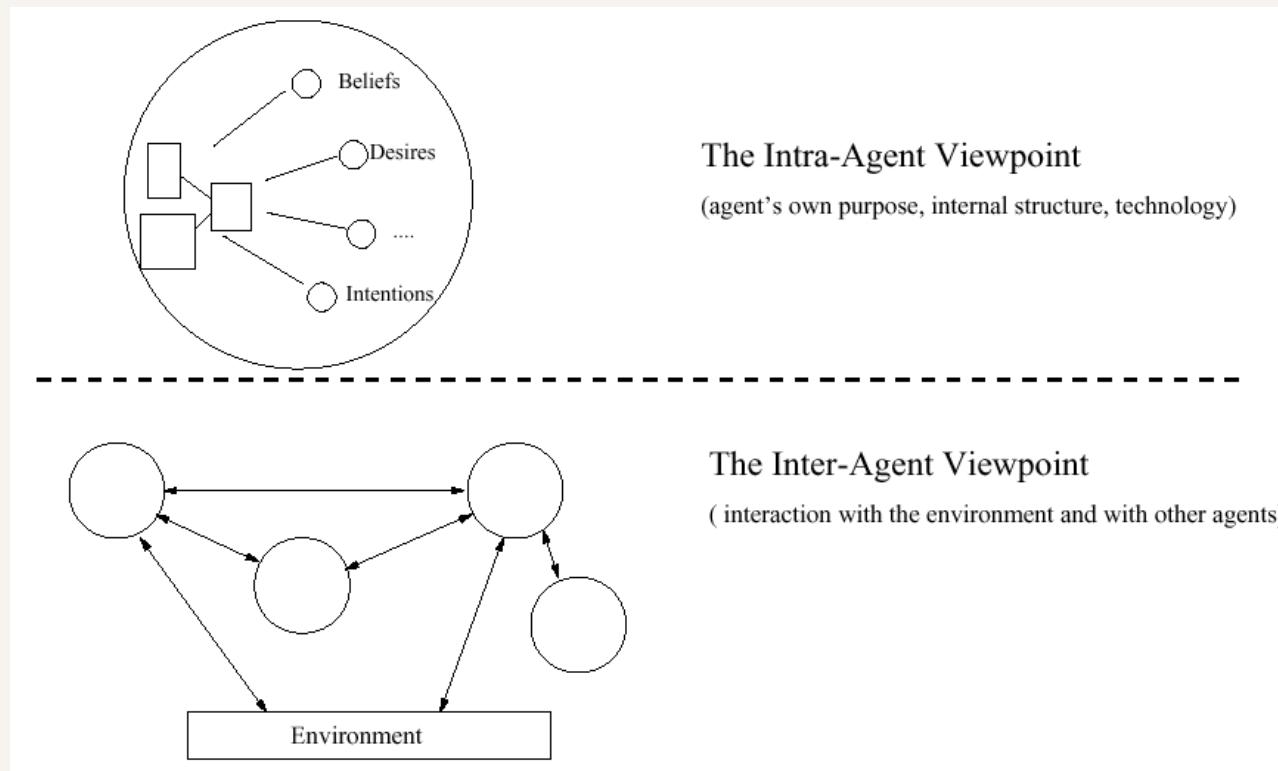
4.1 Introducción

- ✿ Identificar:
 - ✿ los diferentes subsistemas que forman parte del sistema global
 - ✿ las posibles interacciones y dependencias entre ellos



4.1 Introducción

- ✿ A tener en cuenta:
 - ✿ punto de vista interno → un agente
 - ✿ punto de vista externo → varios agentes

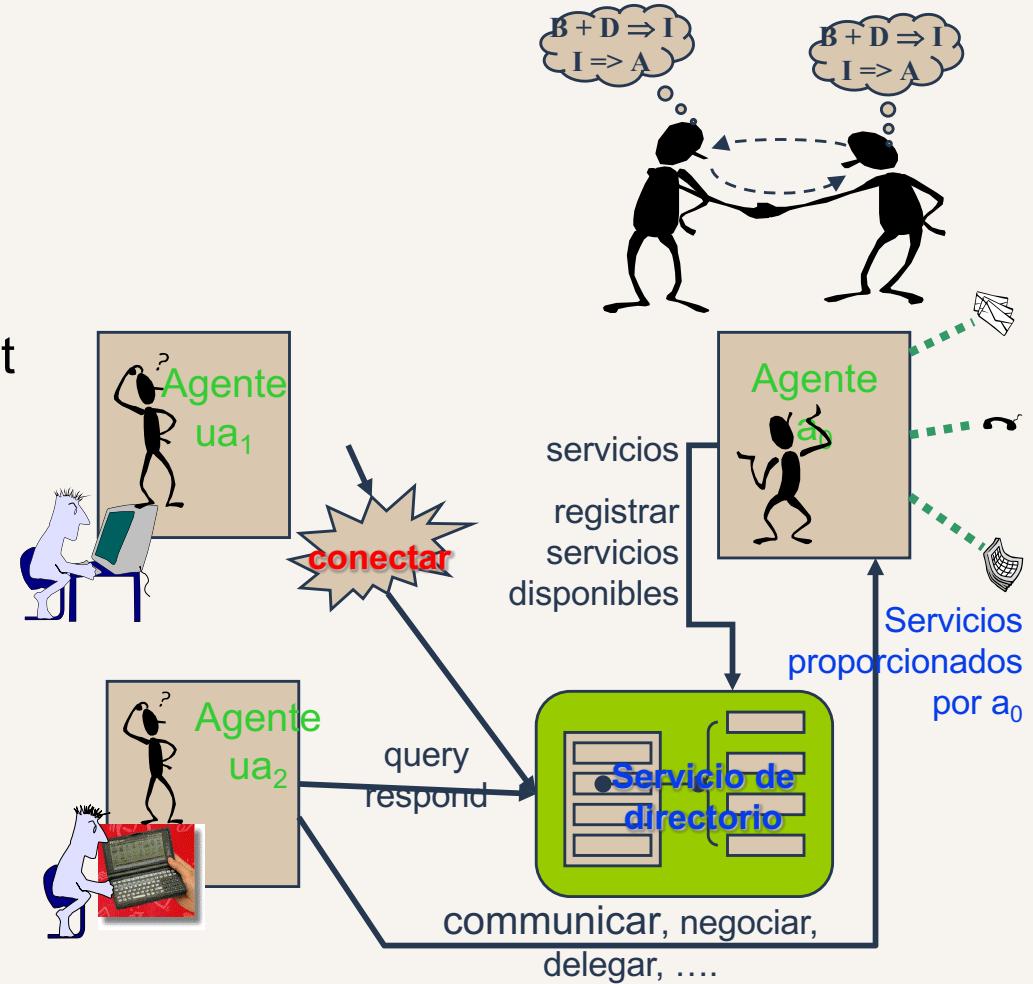


Ventajas

- ✿ **Modularidad:** Se reduce la complejidad al trabajar con unidades más pequeñas, permitiendo una programación más estructurada.
- ✿ **Eficiencia:** La programación distribuida permite repartir las tareas entre los agentes → paralelismo (agentes trabajando en diferentes máquinas).
- ✿ **Fiabilidad:** Que un elemento del sistema deje de funcionar no tiene que significar que el resto también lo tenga que hacer. Se puede conseguir más seguridad replicando servicios críticos y tener así redundancia.
- ✿ **Flexibilidad:** Se pueden añadir y eliminar agentes dinámicamente.

4.1 Introducción

- ❖ Sin estándares, no hay agentes
 - ❖ FIPA (*Foundation for Intelligent Physical Agents*)
- ❖ Sin herramientas sw, no hay agentes
 - ❖ JADE, MAGENTIX2
- ❖ Estándares + Herramientas Sw = agentes?



Sistemas Multiagente

Tema 4: Sistemas Multiagente



Vicent Botti
Universitat Politècnica de València

GTI IA Grupo de Tecnología Informática
Inteligencia Artificial



Asociación Española para
la
Inteligencia Artificial
(AEPIA)

UIMP Universidad Internacional
Menéndez Pelayo

24

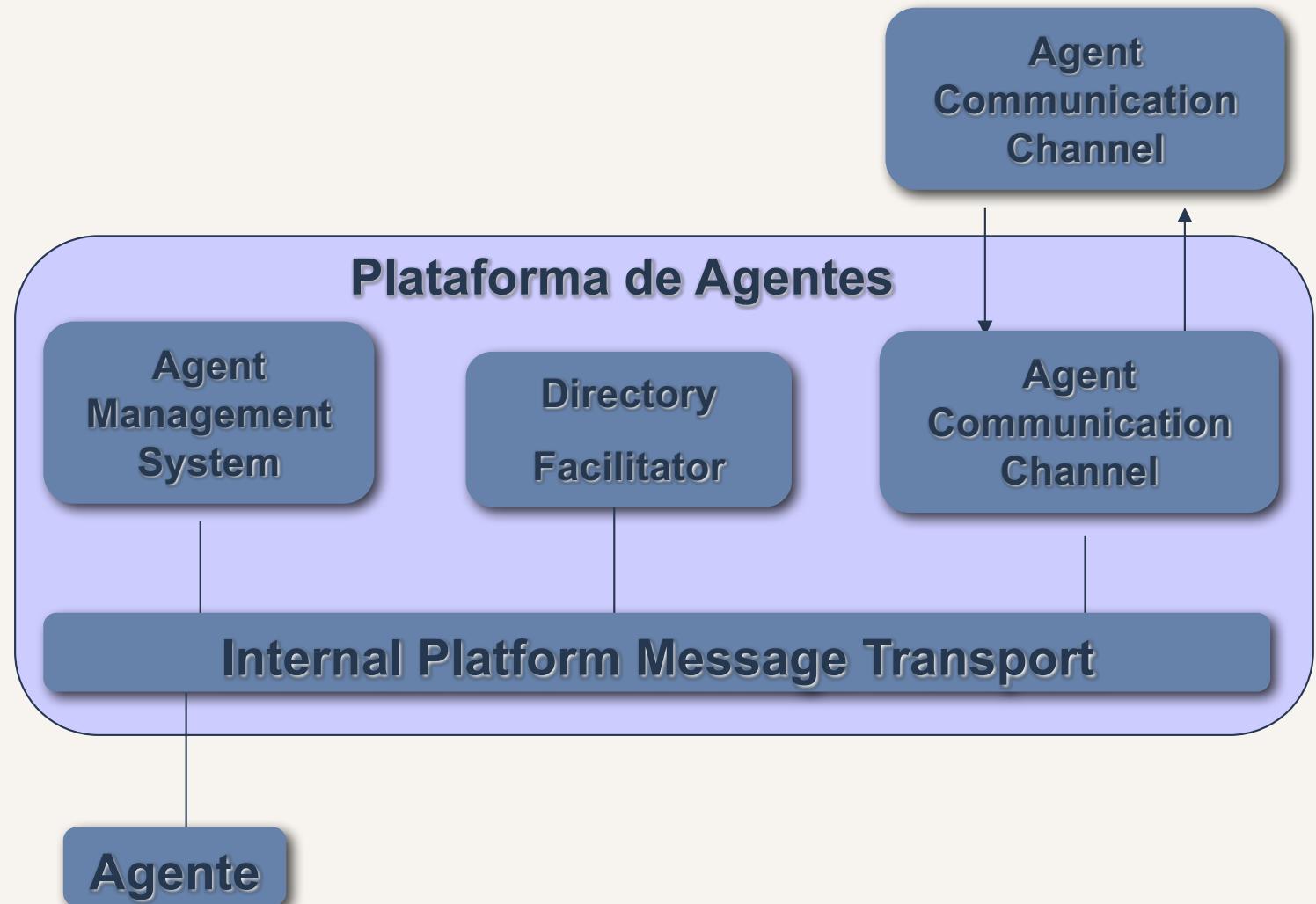
4.2 Modelo Conceptual de Plataforma FIPA

FIPA (Foundation for Intelligent Physical Agents), la organización de estándares para agentes y sistemas multiagente fue oficialmente aceptada por la IEEE el 8 de junio de 2005.

<http://www.fipa.org/>



4.2 Modelo Conceptual de Plataforma FIPA



4.2 Modelo Conceptual de Plataforma FIPA

**Infraestructura en la cual los agentes
pueden ser desarrollados y usados.**

Hw y Sw

Plataforma de Agentes

Agent
Management
System

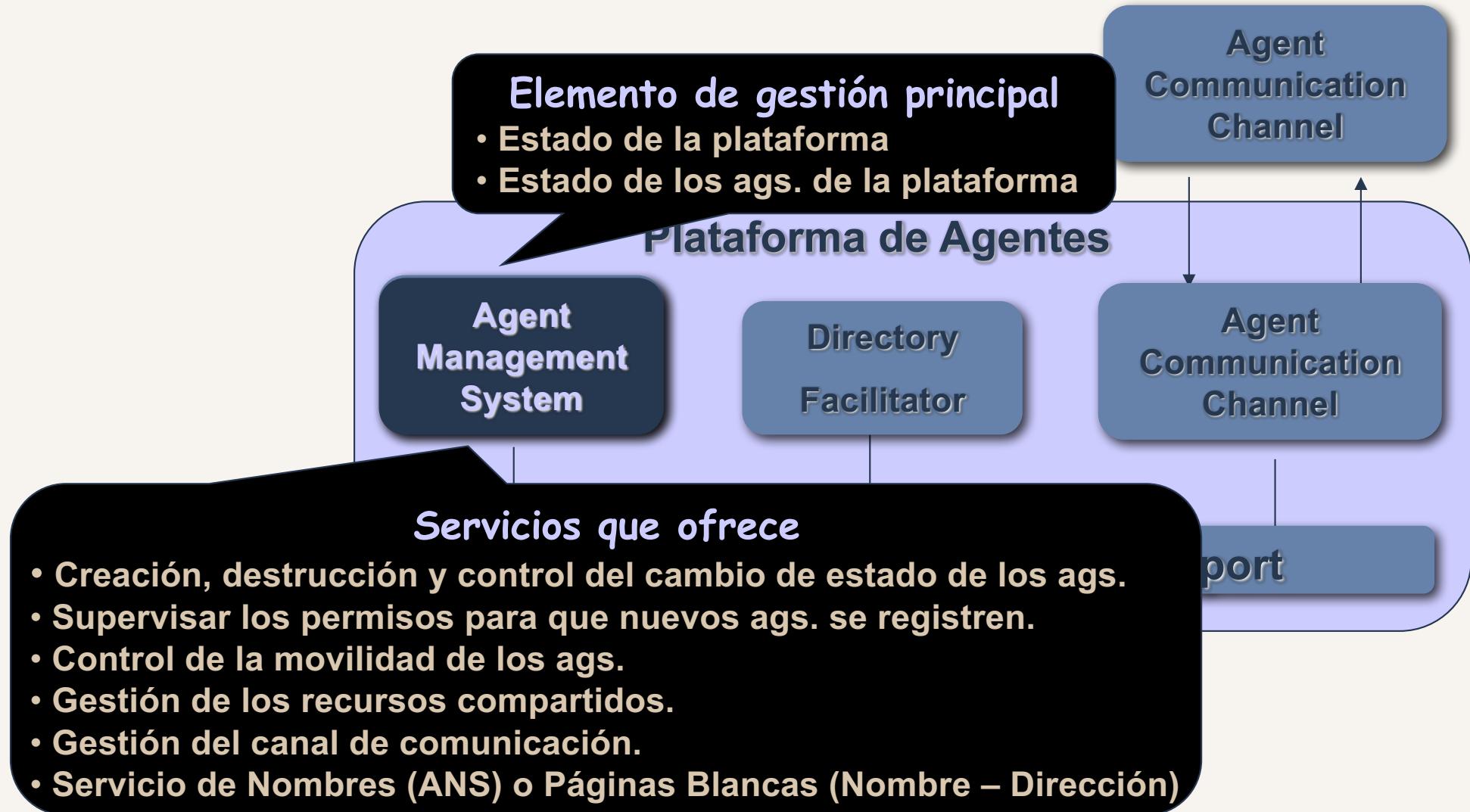
Directory
Facilitator

Agent
Communication
Channel

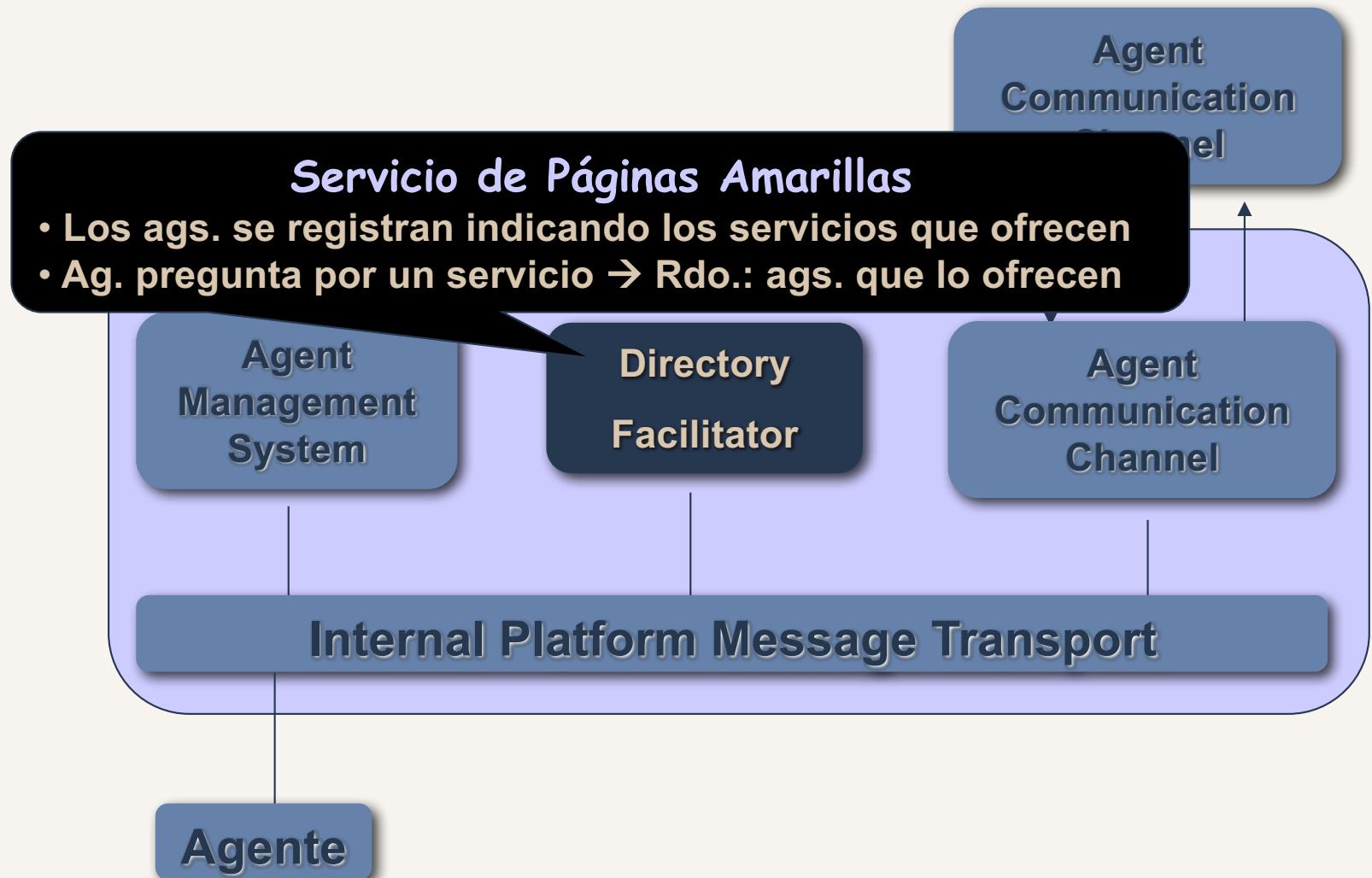
Internal Platform Message Transport

Agente

4.2 Modelo Conceptual de Plataforma FIPA



4.2 Modelo Conceptual de Plataforma FIPA



4.2 Modelo Conceptual de Plataforma FIPA



4.2 Modelo Conceptual de Plataforma FIPA



Sistemas Multiagente

Tema 4: Sistemas Multiagente



Vicent Botti
Universitat Politècnica de València

GTI IA Grupo de Tecnología Informática
Inteligencia Artificial



Asociación Española para
la
Inteligencia Artificial
(AEPIA)

UIMP Universidad Internacional
Menéndez Pelayo

32

4.3 Plataformas de Agentes

¿Cómo ejecutamos un sistema multiagente?

Plataformas de Agentes

¿Qué nos debería ofrecer una plataforma de agentes?

¿Qué plataformas hay existentes?

4.3 Plataformas de Agentes

- Middleware:
 - Software de **conectividad**
 - Ofrece un **conjunto de servicios** que hacen posible el funcionamiento de aplicaciones distribuidas sobre **plataformas heterogéneas**
 - Funciona como una capa de **abstracción de software distribuida**, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red).
 - El Middleware nos **abstira de la complejidad y heterogeneidad de las redes de comunicaciones** subyacentes, así como de los sistemas operativos y lenguajes de programación, **proporcionando una API para facilitar la programación** y manejo de aplicaciones distribuidas.

4.3 Plataformas de Agentes

- Middleware:
 - Dependiendo del problema a resolver y de las funciones necesarias, serán útiles diferentes tipo de servicios de middleware.
- Plataforma:
 - Proporciona un **alto nivel de abstracción** para el desarrollo de determinadas aplicaciones
 - Conjunto de **herramientas** que facilitan el desarrollo
 - **API** adaptada al dominio de las aplicaciones

4.3 Plataformas de Agentes

- ¿Por qué plataformas y middleware para agentes?
 - lenguajes de programación de agentes
 - ‘sistemas operativos’ de agentes
 - Utilización de tecnologías conocidas y que **funcionan**
 - **Interoperabilidad** entre distintos desarrollos de agentes

4.3 Plataformas de Agentes

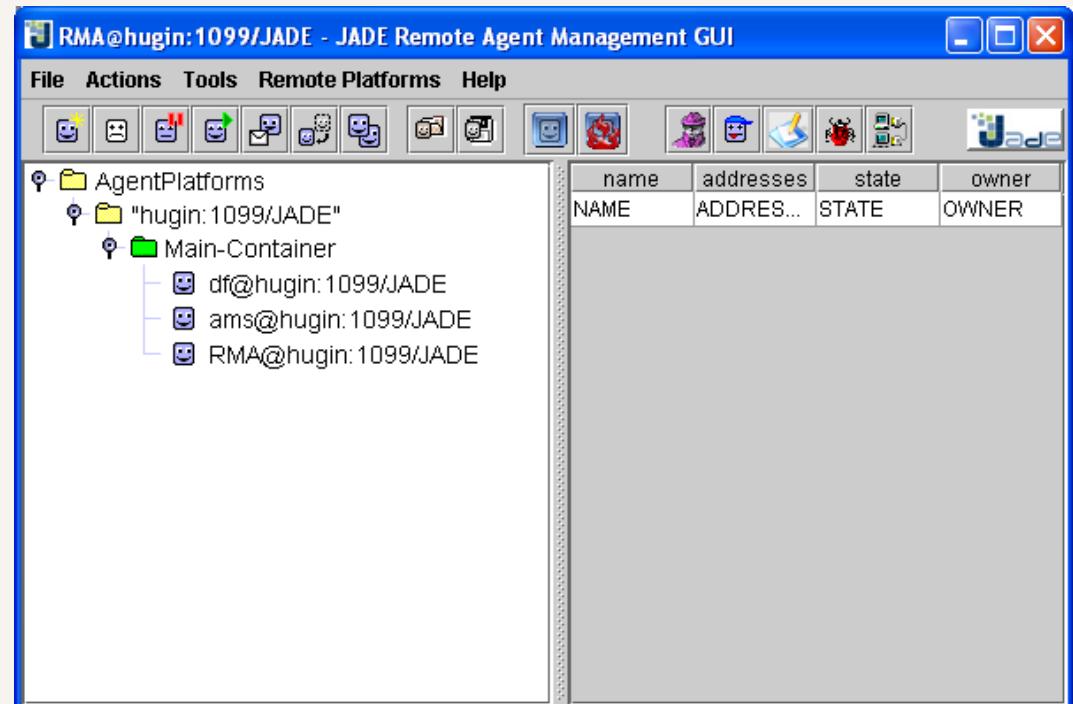
- ❖ A-GLOBE <http://exile.felk.cvut.cz/aglobe>
- ❖ MaDKit <http://www.madkit.org/>
- ❖ JANUS <http://www.janus-project.org>
- ❖ AgentScape <http://www.iids.org/research/aos>
- ❖ AMELI <http://e-institutor.iiia.csic.es/islander/pub/index.html>
- ❖ AGV <http://www.jbtc-agv.com/>
- ❖ JADE <http://jade.cselt.it>
- ❖ JADEX <http://jadex.informatik.uni-hamburg.de>
- ❖ MAGENTIX2 <http://gti-ia.upv.es/sma/tools/magentix2/index.php>

La más usada

- ✿ JADE (Java Agent Development Framework) : Marco de trabajo para el desarrollo de SMA y aplicaciones acorde con el estándar FIPA:
 - ✿ Plataforma para agentes (FIPA-compliant).
 - ✿ Paquete para el desarrollo de agentes en Java.

Java Agent DEvelopment Framework

- ❖ Combinación de dos productos:
 - ❖ Una Plataforma de Agentes FIPA-Compliant
 - ❖ Una herramienta de desarrollo de agentes Java



JADE



Links

<http://jade.cselt.it>

<http://www.fipa.org>

<http://www.hibernate.org>

<http://herzberg.ca.sandia.gov/jess/>

<http://protege.stanford.edu/>

www.swi.psy.uva.nl/usr/aart/beange

<http://jadex.sourceforge.net>



MAGENTIX2

- ❖ Objetivo principal: Ofrecer soporte al desarrollo de SMA abiertos
 - Agentes programados en cualquier lenguaje de programación
 - Soporte para múltiples arquitecturas de agentes
 - Interacción entre agentes transparente
 - Soporte para protocolos de interacción dinámicos
 - **Agentes BDI**
 - Organizaciones de agentes

Communication
Infrastructure

Tracing

Security

Conversational
Agents

THOMAS
Framework

Magentix 2

Development

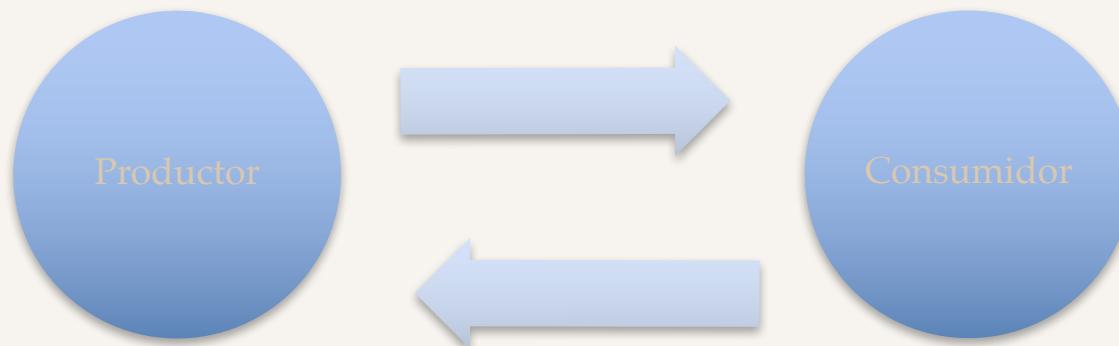
Installation

Jason and J-
Moise

MAGENTIX2

- ✿ Un agente Magentix2 tiene tres métodos principales: *init*, *execute* y *finalize*.
- ✿ Estos métodos son ejecutados en ese orden.
- ✿ *Init*: el código se ejecuta al principio de la ejecución del agente.
- ✿ *Execute*: es el método que contiene el código principal del agente
- ✿ *Finalize*: este método se ejecuta justo antes que el agente finalice y sea destruido.
- ✿ Sólo es obligatorio incluir código en el método *execute*.

Ejemplo:



Magentix2

Sistemas Multiagente

Tema 4: Capacidad Social



Vicent Botti
Universitat Politècnica de València

GTI IA Grupo de Tecnología Informática
Inteligencia Artificial



Asociación Española para
la
Inteligencia Artificial
(AEPIA)

UIMP Universidad Internacional
Menéndez Pelayo