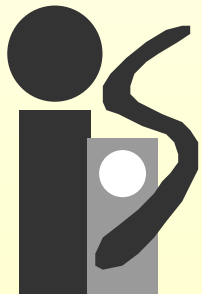
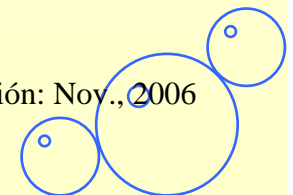
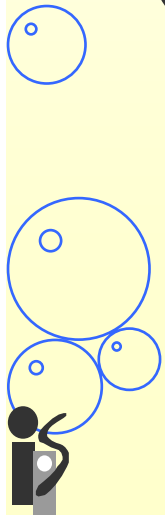


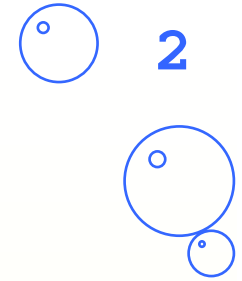
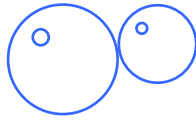
Introducción a la Tecnología de Agentes



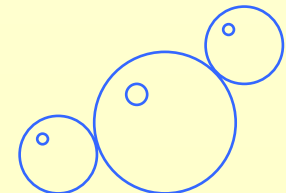
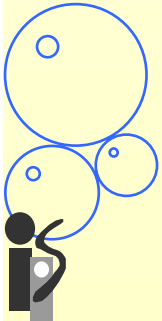
J.A. Bañares Bañares

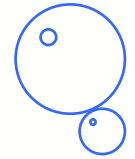
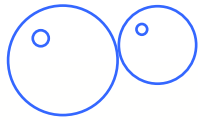
Departamento de Informática e Ingeniería de Sistemas
C.P.S. Universidad de Zaragoza



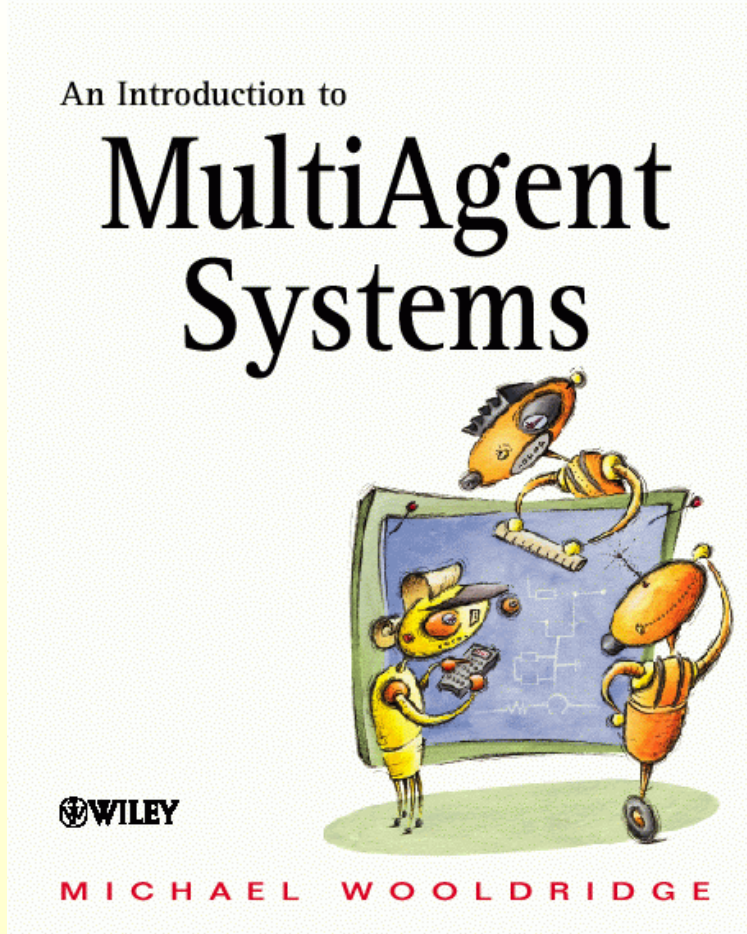


-
- **Esta presentación ha sido elaborada a partir del material docente del Curso de doctorado**
 - Agentes Inteligentes de la Universidad Complutense de Madrid del profesor Juan Pavón Mestras
 - <http://www.fdi.ucm.es/profesor/jpavon/doctorado/>
 - Material en Agentcities.es
 - <http://grusma2.etse.urv.es/AgCitES/index.htm>

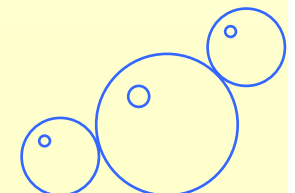
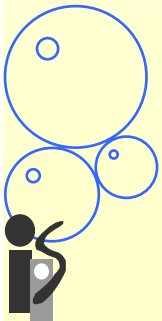


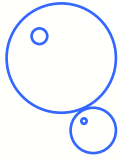
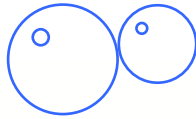


Libros básicos sobre agentes

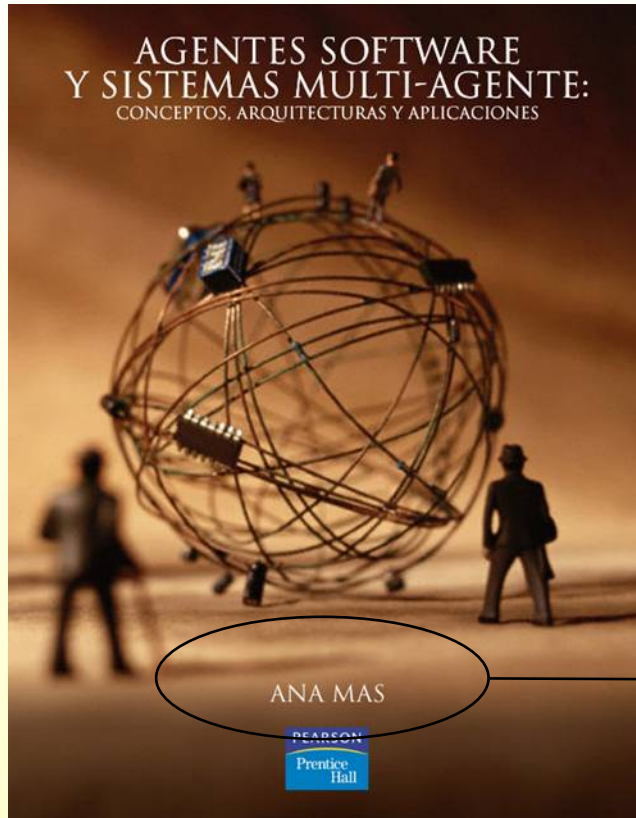


*Dof#qvroxfwiof#z#PxabjhqW/vhpo ve|# lfkdh#
Zrrqubjhl\$xedkng b#heudu| 5335#e|#tkof#Zb|)#
Vrcv +Fkfkhwu/#qjdog, #VEQ#:# :47<9<4 [1#
673ss#Bssur{#bfoghveledrjudsklfduhnhqfhw dog
bgh{#*



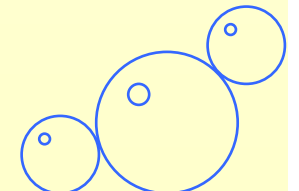
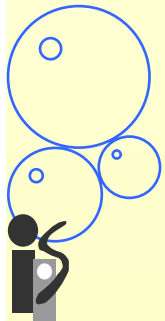


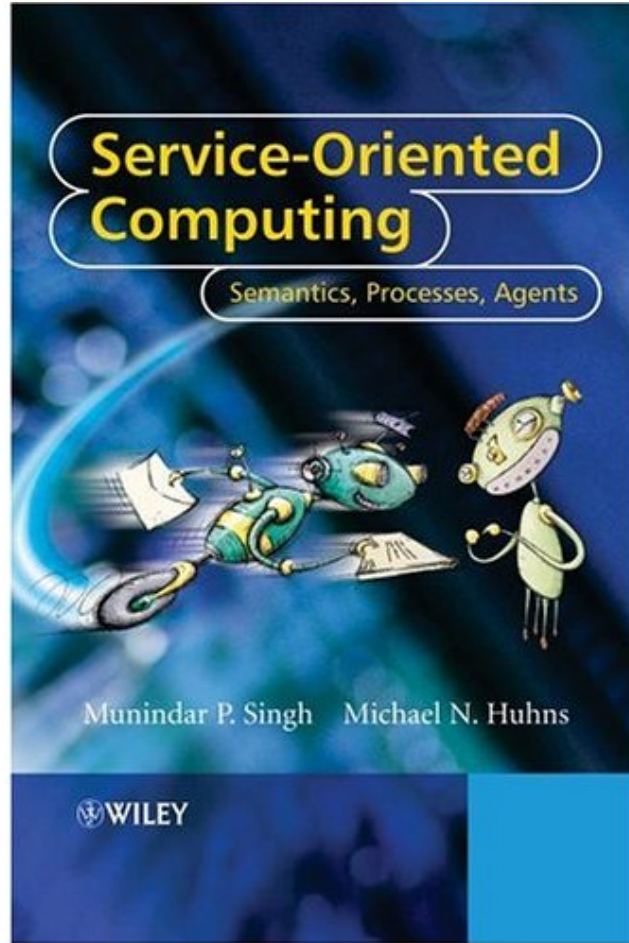
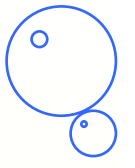
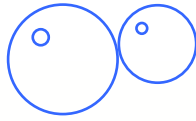
Libros básicos sobre agentes



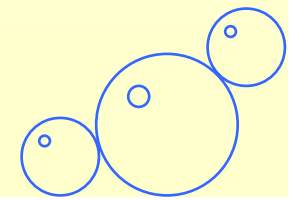
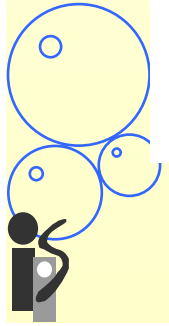
Wwcr=Djhqhv#vzduh##vwhp d#pxodjhqhv
 Vxwcr=#rqlfhsrv/utxwhfwudv|#sdfdrqhvh##VEQ#
 ;705380769:08##Dwru#kdj#Edy&q/Mv#DfS+h)#gho#
 Fx)#Prdpl# S<jv#585##SfS-#;#ixur#sur{

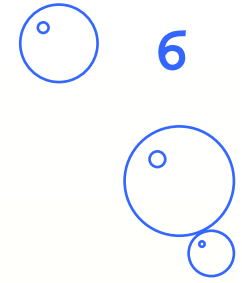
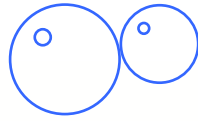
Dhzh dxwkrul| rqlpxodjhqhv|whpv





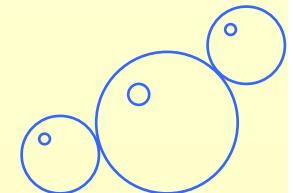
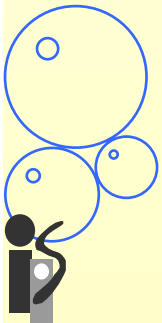
VnyfnRuhqng Fp sxbj #hp dqwv/Surfwhv/#jhw -Kdugryu,#
e|#xq|pdu.S#vbjk/#fkohdQ#kxqv

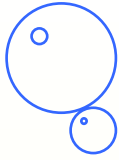
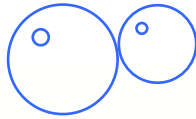




Objetivo

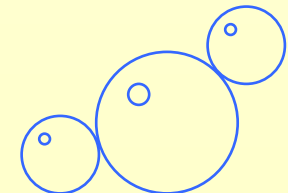
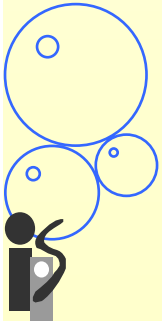
- **Objetivo:**
 - ¿Qué son?
 - ¿Por qué otro paradigma?
 - ¿Otra moda tecnológica?
 - ¿Qué hay de nuevo?
 - ¿Cómo se construyen?
 - ¿Por dónde empezar?
 - ¿Hasta dónde podemos llegar?
 - ¿Nuestro *álder-ego* en la Red?

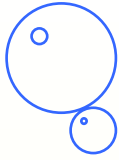
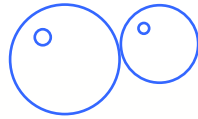




Índice

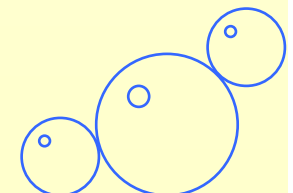
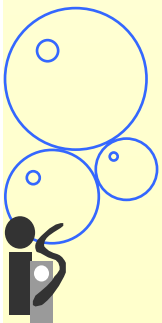
- **Definición Agente software y Sistemas Multi-agente**
- **Características de los agentes**
 - Autonomía
 - Inteligencia
 - Sistemas multiagentes/ Arquitecturas cooperativa
 - Habilidades sociales
 - Movilidad
- **Clases/Categorías de agentes**
- **Ejemplos**
- **Agentes como componentes software vs Objetos/SE**
- **Comunicación entre agentes**
 - Lenguajes de Comunicación (KIF, KQML, Speech-Acts)
 - Protocolos
- **FIPA (Foundation for Intelligent Physical Agents)**

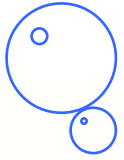
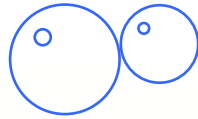




Agentes Software

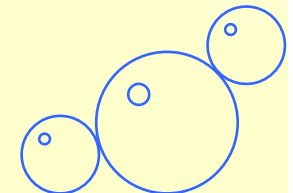
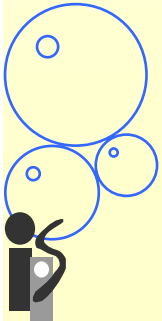
- Los agentes software surgen dentro del campo de la Inteligencia Artificial y, a partir de los trabajos desarrollados en el área de la *Inteligencia Artificial Distribuida* (DAI), surge el concepto de sistemas multiagente
- La investigación inicial progresa hacia la madurez. Surge la *Programación Orientada a Agentes* y los *Lenguajes de Comunicación de Agentes*
- Posteriormente este concepto se extiende al resto de la Ingeniería del software, planteándose hoy en día la *Ingeniería del Software Orientada a Agentes*

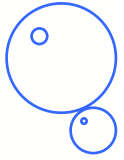
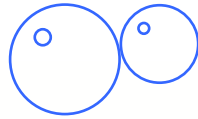




Definición de agente Software

- No hay definición universalmente aceptada
- Varios tipos de definiciones
 - Definiciones cualitativas
 - Basadas en propiedades o en habilidades del agente, e.g. definición de Wooldridge & Jennings
 - Definiciones operacionales
 - Test de Huhns- Singh
 - Clasificaciones





Definiciones

Agente (Diccionario RAE):

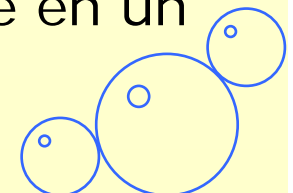
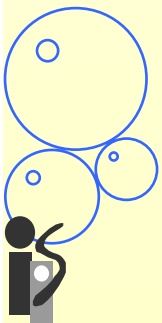
- Que obra o tiene virtud de obrar
 - **El que realiza una acción**
 - **Persona o cosa que produce un efecto**
- Persona que obra con poder de otra
 - **El que actúa en representación de otro (agente artístico, comercial, inmobiliario, de seguros, de bolsa, etc)**
- Persona que tiene a su cargo una agencia para gestionar asuntos ajenos o prestar determinados servicios

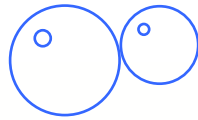
Agentes software (softbots):

- Aplicaciones informáticas con capacidad para *decidir* cómo deben actuar para alcanzar sus objetivos

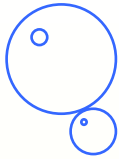
Agentes inteligentes:

- Agentes software que pueden funcionar fiablemente en un entorno rápidamente cambiante e impredecible



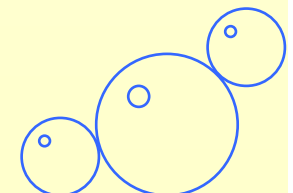
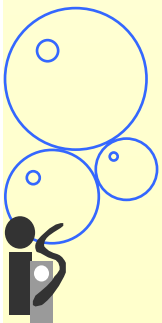


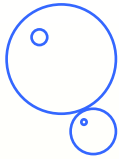
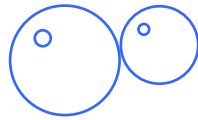
El papel de los agentes en la IA



Proyectando las tendencias actuales en el futuro, pienso que habrá un nuevo énfasis sobre sistemas autónomos- robots y softbots. Softbots son agentes software que navegan por Internet, buscando información que pueda ser interesante para sus usuarios. La presión para mejorar las capacidades de los robots y los agentes software motivarán y guiarán la investigación en IA durante los próximos años."

(Artificial Intelligence, A new Synthesis. Nilsson 1998)



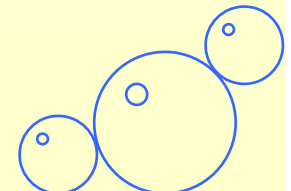
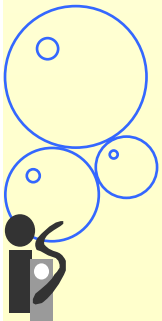


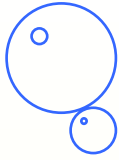
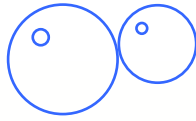
El papel de los agentes en la IA

Los agentes son el elemento unificador de todo el contenido de libros de texto actuales de introducción a la IA

La tarea de la IA es explicar y construir agentes que reciben percepciones del ambiente y proceden a ejecutar acciones.

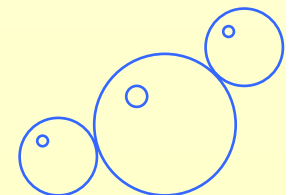
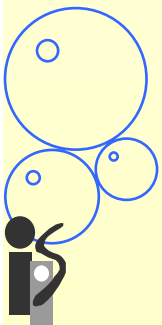
(Artificial Intelligence, A modern approach. Stuart Russel & Peter Norvig 1996)

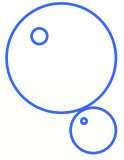
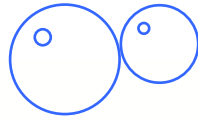




Definiciones de Wooldridge

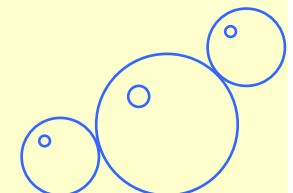
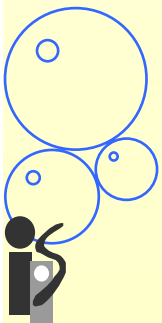
- **Un agente** es un software capaz de acciones *independientes* en representación de su propietario o usuario (comprendiendo las necesidades que tienen que ser satisfechas para satisfacer sus objetivos de diseño, en lugar de recibir instrucciones constantemente).
- **Un sistema multiagente** consiste en un conjunto de agentes que *interaccionan* entre ellos. En el caso más general en representación de usuarios con diferentes motivaciones y objetivos. Par conseguir interaccionar con éxito deben *cooperar*, *coordinarse* y *negociar*, como hace la gente.

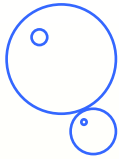
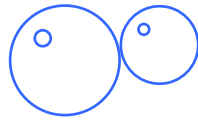




Diseño de Agentes, Diseño de sociedades

- **Se cubren dos aspectos claves:**
 - ¿Cómo construir agentes independientes, capaces de acción autónoma, que puedan llevar a cabo tareas que les encarguemos?
 - ¿Cómo construir agentes capaces de interaccionar (cooperar, coordinar, negociar) con otros agentes para llevar a cabo las tareas que se les han encargado, especialmente cuando los otros agentes no comparten los mismos intereses (objetivos)?
 - El primer aspecto es *diseño de agentes*, el segundo *diseño de sociedades* (micro/ macro)



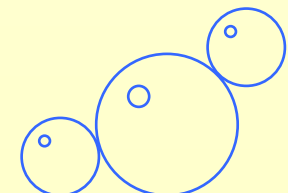
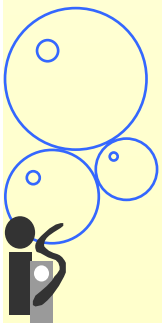


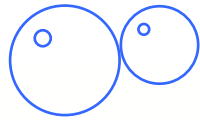
I Primera característica

- Los agentes son:

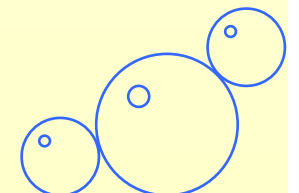
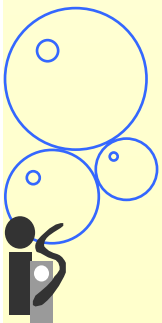
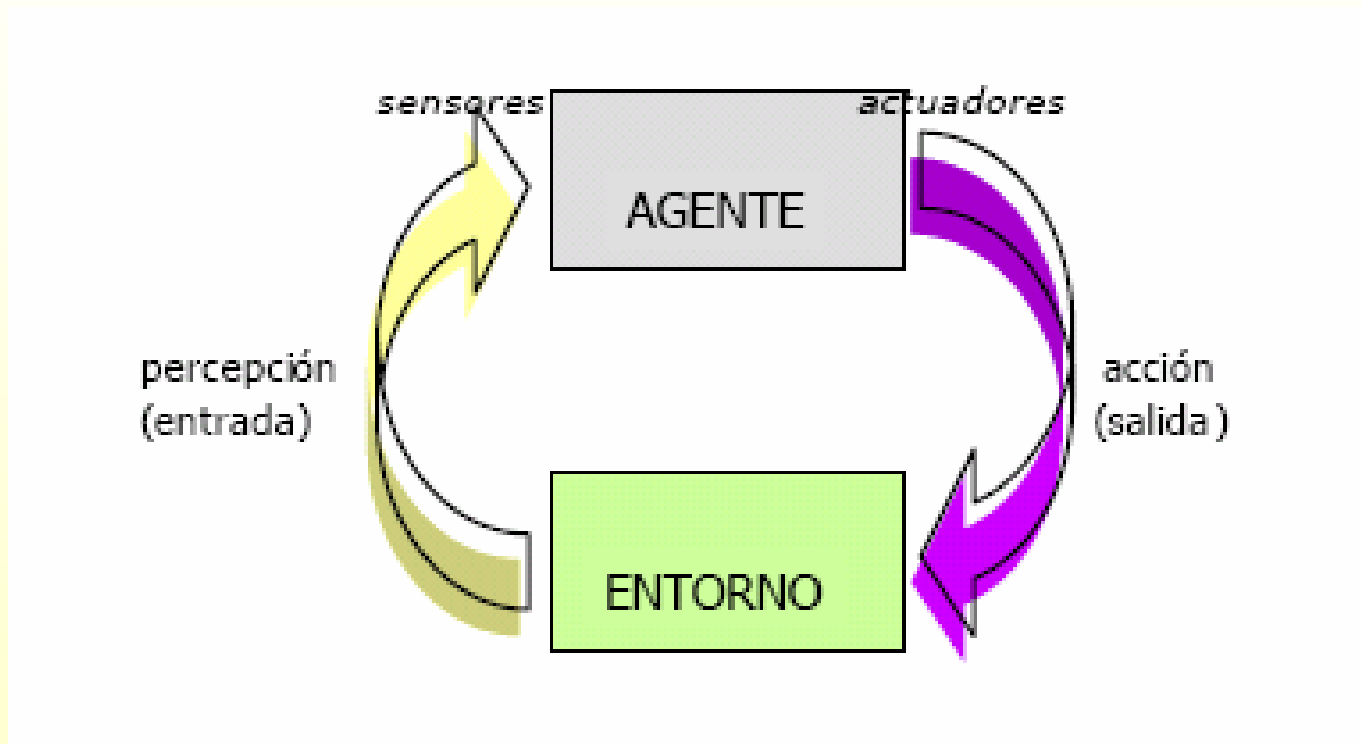
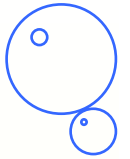
Entidades autónomas

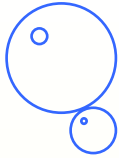
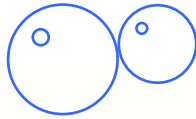
- Autonomía
 - Pueden trabajar sin la intervención directa del usuario y tienen cierto control sobre sus acciones y estado interno
- Reactividad
 - Pueden percibir su *entorno* (que puede ser el mundo físico, un usuario detrás de una interfaz gráfica o vocal, aplicaciones en la red, u otros agentes) y responder oportunamente a cambios que se produzcan en el mismo
- Iniciativa
 - El comportamiento de los agentes está determinado por los *objetivos* (metas) que persiguen y por tanto pueden producir acciones no sólo como respuesta al entorno





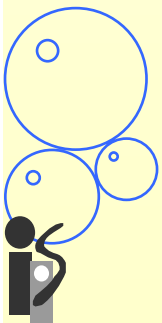
Agentes como entidades autónomas que interaccionan con el entorno

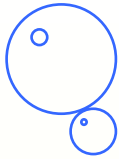
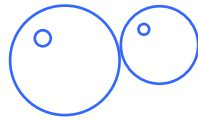




Relación del agente con el entorno

- En la mayor parte de los dominios el agente sólo tendrá control parcial del entorno
- Una misma acción realizada por el agente puede tener efectos muy distintos
 - **En general los entornos son no-deterministas**
 - **Un agente debe estar preparado para fallar o para la incertidumbre de no saber si ha tenido éxito o no**
- Un agente dispone de un repertorio de acciones disponibles con sus correspondientes precondiciones
 - **Modelo Agensheet**
- El principal problema al que se enfrenta un agente es decidir qué acción realizar para alcanzar sus objetivos de diseño
- Las arquitecturas de agentes podrían verse como arquitecturas software para sistemas de toma de decisiones empotrados en un entorno



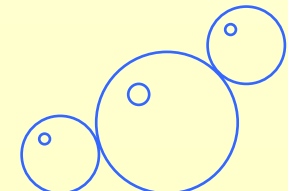
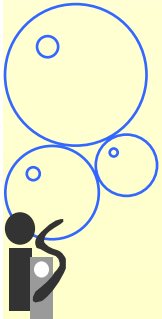


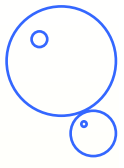
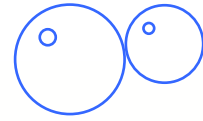
II Segunda Característica

- Los agentes tienen:

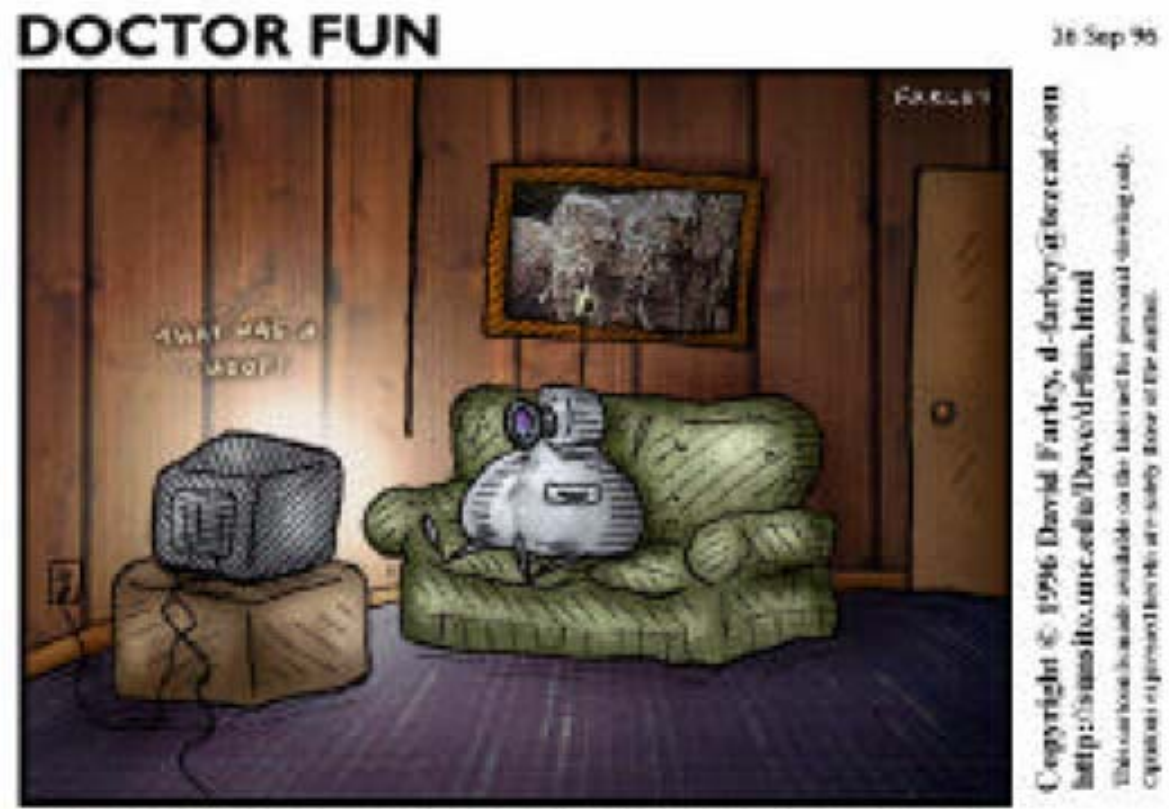
Inteligencia

- Razonamiento
 - Un agente puede decidir:
 - **qué objetivo perseguir o a qué evento reaccionar**
 - **cómo actuar para conseguir un objetivo**
 - **o suspender o abandonar un objetivo para dedicarse a otro**
- Aprendizaje
 - El agente puede adaptarse progresivamente a cambios en entornos dinámicos mediante técnicas de aprendizaje

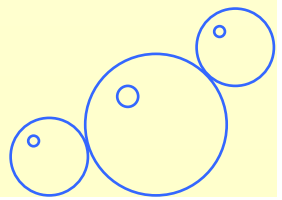
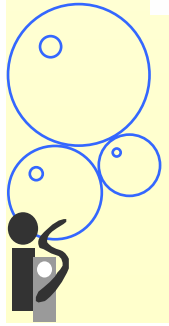


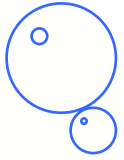
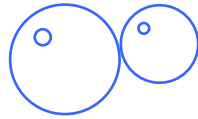


Agentes Inteligentes para delegar



Hq#b#ixur#e#v#j#hqw#v#q#w#j#hqw#v#g#h#f#d#x#q#x#h#p#s#r#b#
 y#u#o#h#b#y#v#q#s#r#u#r#v#w#v#l#l



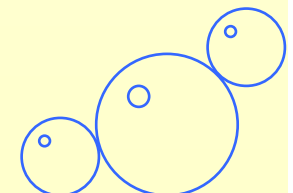
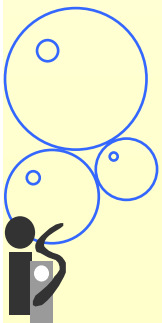


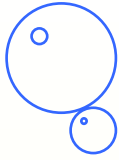
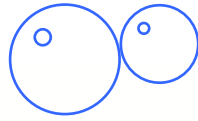
Agentes inteligentes

Principio de Racionalidad (Allen Newell (1982). *The knowledge level*. Artificial Intelligence **18**: 87-127):

If an agent has knowledge that one of its actions will lead to one of its goals, then the agent will select that action

- Indica que hay una conexión entre objetivos y comportamiento, por medio del conocimiento del que dispone el agente ... lo que no implica que el agente tomará la mejor decisión

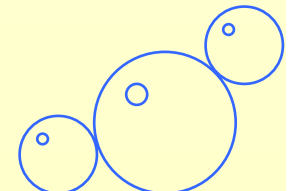
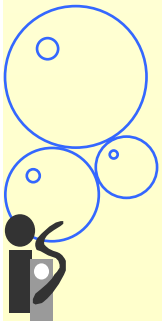


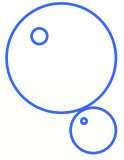
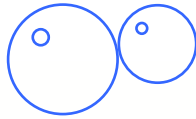


Agentes: Inteligencia

- Técnicas para implementar la racionalidad
 - Representación del Conocimiento
 - Razonamiento simbólico
 - Planificación
 - Satisfacción de Restricciones
 - Aprendizaje Automático
 - Razonamiento bajo incertidumbre:
 - Sistemas de mantenimiento de la verdad
 - Inferencia Estadística

⇒ **!! Sistemas Basados en el Conocimiento!!**



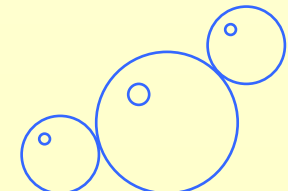
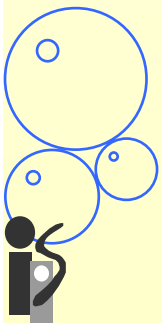


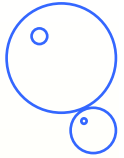
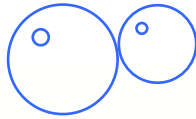
III Característica de los agentes

- Los agentes no actúan solos, sino como:

Sistemas Multi-Agentes

- Resolución de problemas mediante la estrategia *divide y vencerás*
 - Reparto de responsabilidades-Identificar Roles
- *Heterogeneidad*
 - Especialización.
 - Marco para la integración de Bases de conocimiento heterogéneas especializadas en tareas
 - Para abordar la interoperabilidad surge el rol de *Middle agents*, *Mediators*, *Avatars*, *interface Agents*,...
- *Concurrencia y Distribución*
 - Flexibilidad, escalabilidad, tolerancia a fallos, gestión de recursos
 - Distribución del conocimiento
 - Diferentes arquitecturas de control, con coordinador, sin coordinador, ...



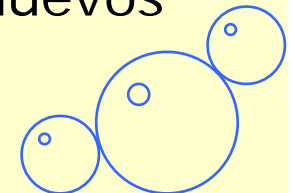
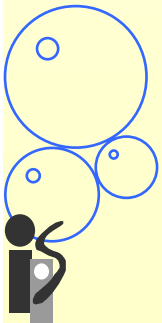


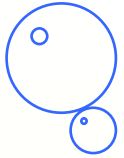
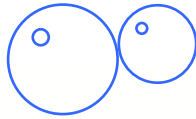
Test de Huhns-Singh

- *A system containing one or more reputed agents should change substantively if another of the reputed agents is added to the system*

Hunhns, M.; Singh, M.P. *A Multiagent Treatment of Agenthood*. en *Applied Artificial Intelligence: An International Journal*, Volume 13, No. 1-2, January - March 1999, pp. 3-10

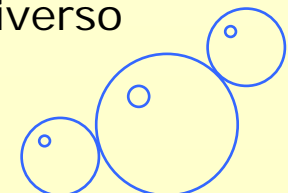
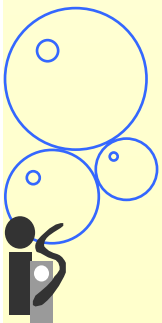
- El test requiere ciertas condiciones:
 - El entorno del agente no es estático (i.e. pueden acaecer eventos)
 - El entorno es observable suficientemente (i.e. no tiene que ser completamente observable)
 - El tipo de los agentes que se incorporan es alguno ya existente en el sistema (aunque puede considerarse que haya varios tipos de agentes).
- Comportamiento emergente
- Mejora del rendimiento del SMA (esto implica que los agentes han de ser conscientes de la aparición de nuevos agentes)

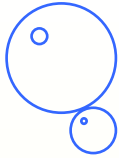
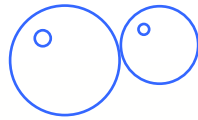




Definición de SMA (Ferber, 1999)

1. Un entorno
2. Un conjunto de objetos
 - Los objetos se encuentran integrados con el entorno, i.e. es posible en un momento dado asociar uno de estos objetos con un lugar en el entorno
 - Los objetos son pasivos, pueden ser percibidos, creados, destruidos y modificados por agentes
3. Un conjunto de agentes
 - Objetos especiales que representan las entidades activas del sistema
4. Un conjunto de relaciones que unen objetos y agentes
5. Un conjunto de operaciones
 - hacen posible que los agentes perciban, produzcan, consuman, transformen y manipulen objetos
6. Operadores que representan la aplicación de operaciones sobre el mundo y la reacción de éste al ser alterado
 - Estos operadores se pueden entender como las leyes del universo



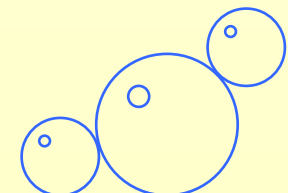
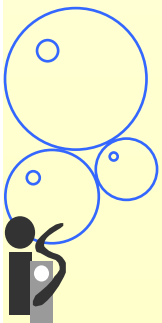


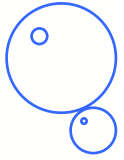
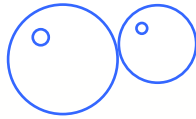
IV Cuarta característica

- Los agentes tienen:

Habilidad Social

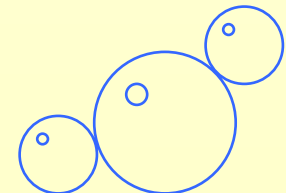
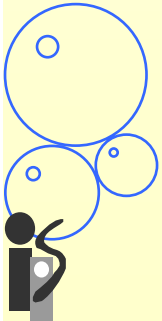
- *Interacción*
Diálogo
- *Delegación*
Asignar la realización de tareas
- *Cooperación*
Trabajo en común para lograr un objetivo común
- *Coordinación*
Organizar el proceso de solución del problema de forma que se eviten interacciones nocivas y que se exploten las beneficiosas
- *Negociación*
Formular un acuerdo que sea aceptable por todas las partes implicadas.

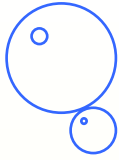
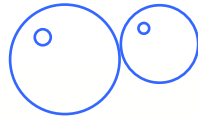




Comunicación entre Agentes

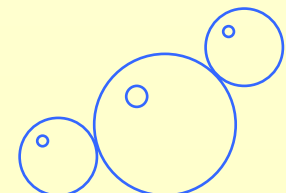
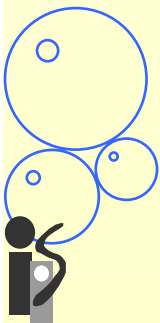
- Lenguajes de comunicación de agentes
 - **KQML**
 - **FIPA ACL**
 - **Basado en la teoría de actos del habla**
- Algunos autores consideran que la capacidad de hablar un lenguaje de agentes es suficiente para considerar a un software como agente





Conversaciones entre agentes

- **Conversación**
 - Intercambio de mensajes
- **Protocolo**
 - Conversación válida
- **Se definen protocolos de negociación, subasta, ... que intercambian mensajes en un lenguaje común.**



¿Influyen los agentes en otras arquitecturas software?

- **Tendencias hacia la interoperabilidad de aplicaciones**

Generación	Primera	Segunda	Tercera	Cuarta
<i>Comunicación</i>	TCP/IP	CORBA	HTTP	Mensajes
<i>Información</i>	SQL	XML	RDF	OWL
<i>Aplicación</i>	RPC	EDI	SOAP	Protocolos
<i>Configuración</i>	Hard-coded	Directorios	UDDI	Selección

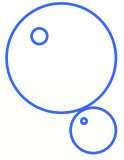
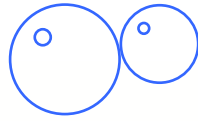
+Vbjk/#xkqv 5338,

¿Influyen los agentes en otras arquitecturas software?

- Tendencias hacia la interoperabilidad de aplicaciones

Generación	Primera	Segunda	Tercera	Cuarta
<i>Comunicación</i>	TCP/IP	CORBA	HTTP	Mensajes
<i>Información</i>	SQL	ORBBus	XMLBus ESBus	RDF/OWL
<i>Aplicación</i>	RPC	EDI/Modelos Dominio	SOAP	Protocolos
<i>Configuración</i>	Hard-coded	Directorios	UDDI	Selección

+Yhuv&op#hsdlwgh#Vbjk/#Kxqv 5338,

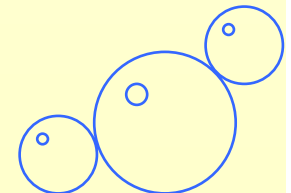
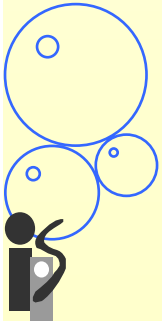


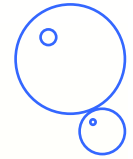
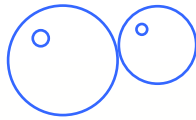
V característica

- Los agentes pueden tener:

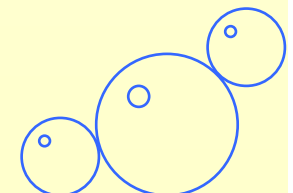
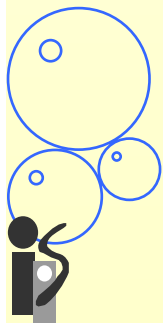
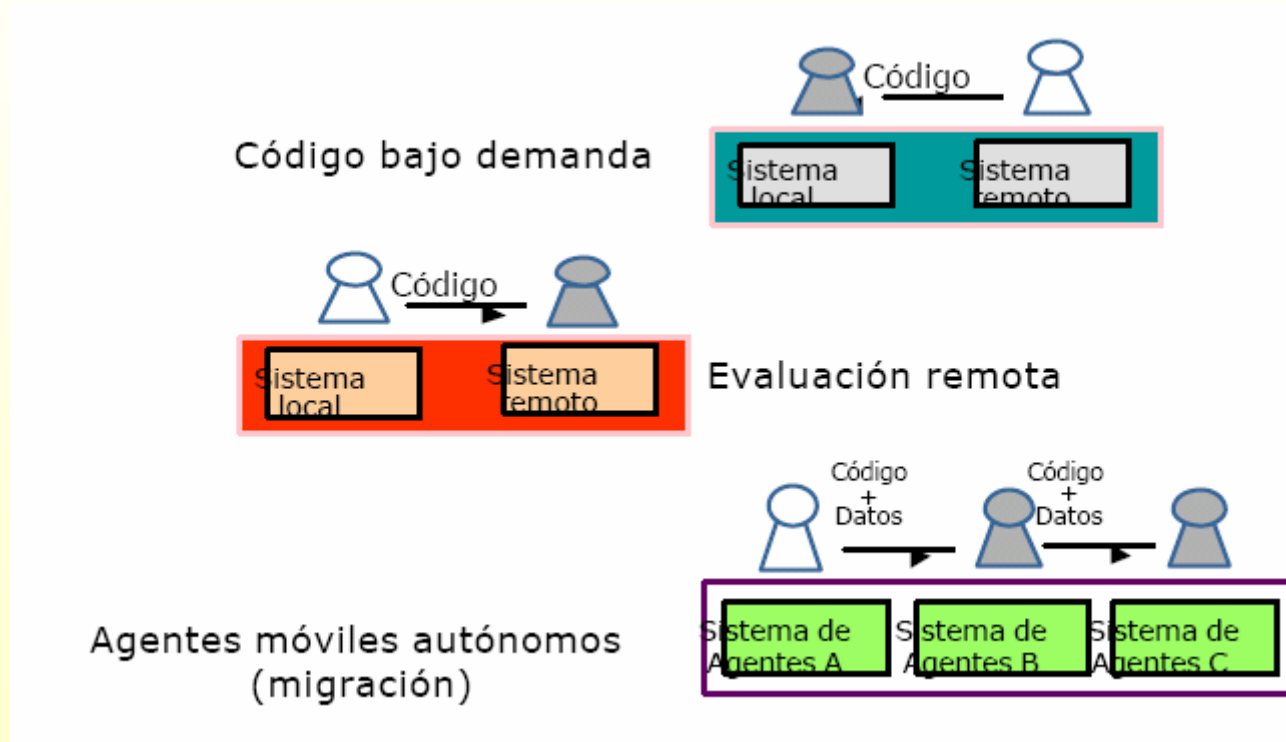
Movilidad

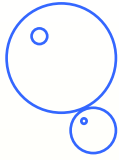
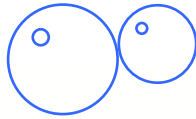
- Agentes móviles
 - Capacidad de Migrar de un nodo a otro en una red preservando su estado en los saltos entre nodos
- Multi-acceso y multi-modal
 - Navegador
 - Email
 - Servicio vocal



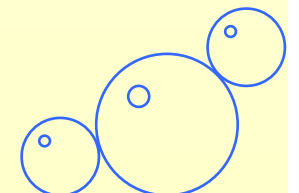
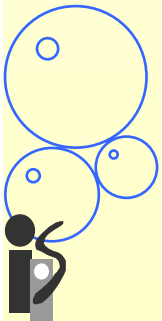
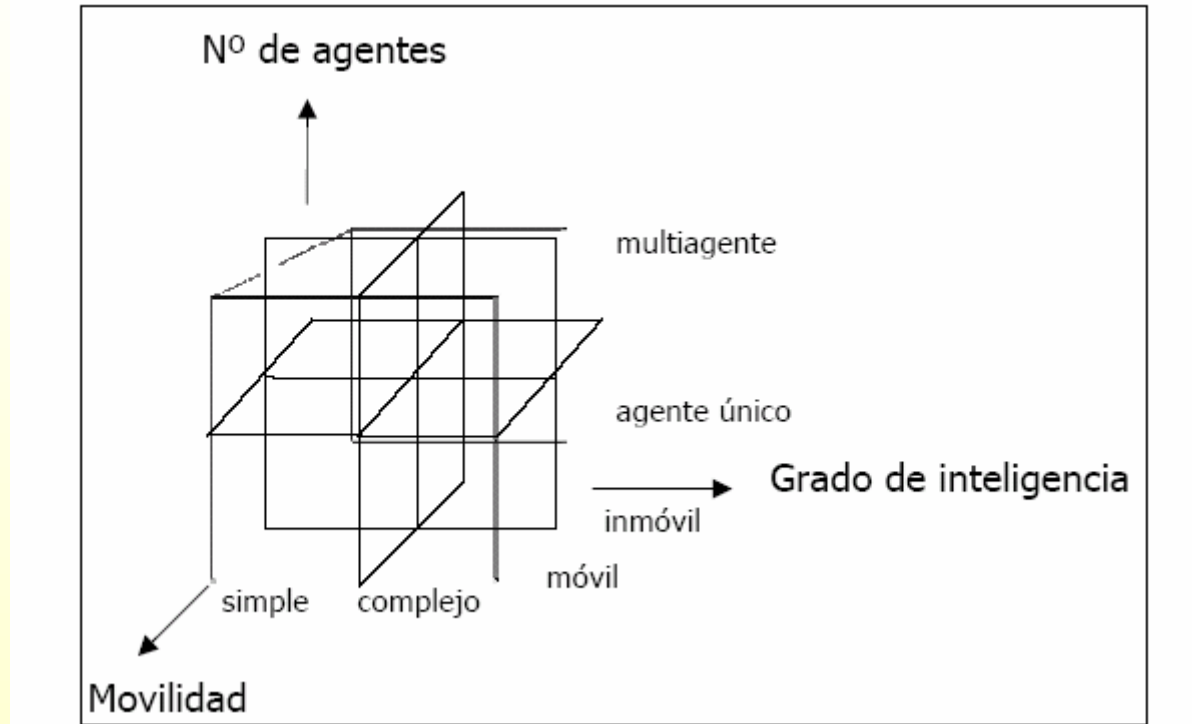


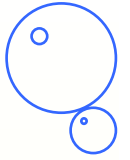
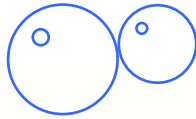
Agentes móviles



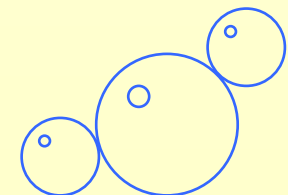
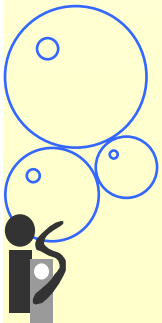


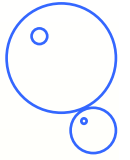
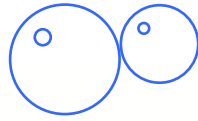
Clasificación de Agentes (M. Wooldridge, 1999)





Categoría de Agentes (Zarnekow, 1998)

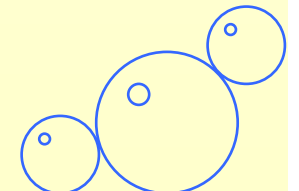
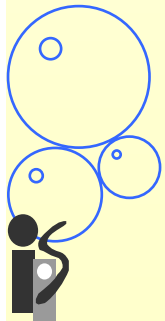
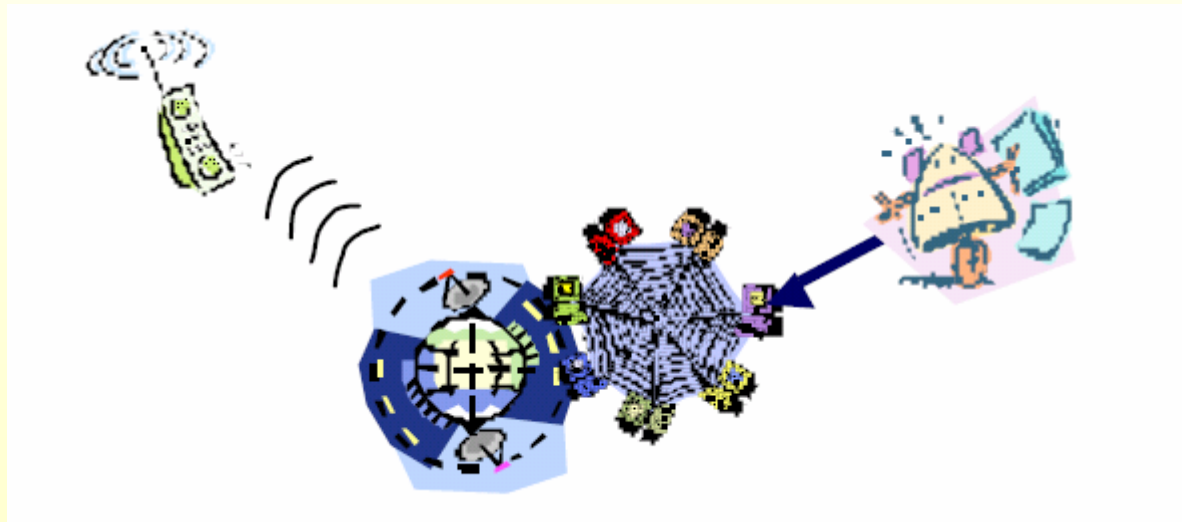


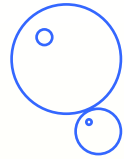
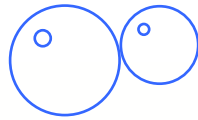


Ejemplo de funcionamiento de SMA

Ejemplo:

Una conversación entre Juan (residente en Madrid) y Carmen (residente en Sevilla) para decidir una reunión de trabajo.





Ejemplo de funcionamiento de SMA

Juan

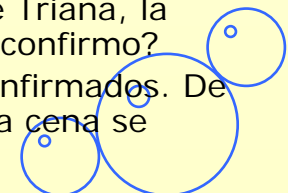
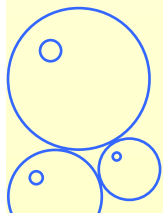
- Hola Carmen! Precisamente estaba pensando llamarte para que preparemos la presentación de nuestra colección de invierno
- (escucha y asiente)
- Está bien, podemos quedar entonces la semana que viene, preguntemos a nuestras agendas...
- Asistente?

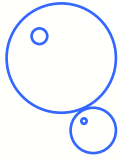
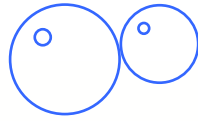
- Bueno Carmen, pues nos vemos el próximo miércoles,
- Hasta luego encanto.
- (cuelga el teléfono)
- Asistente, ¿me organizas el viaje?

- Sí, perfecto
- Sí, muy bien.
- Muchas gracias, asistente.

Asistente de Juan

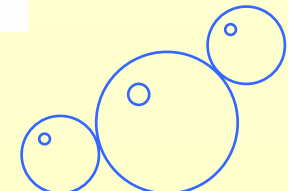
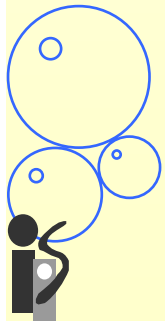
- Buenos días Juan.
De acuerdo con el asistente personal de Carmen, el mejor día para reunirnos es el miércoles 10 de abril en el despacho de Carmen.
- Sí Juan.
- Te propongo salir en el AVE de Madrid el miércoles 10 de abril de 2003 a las 8 de la mañana, llegada a Sevilla 11.30, en clase preferente para que puedas desayunar bien. Vuelta, salida el jueves 11 de abril de 2003 a las 11 de la mañana, llegada a Madrid a las 13.30. en clase turista Lo confirmo?
- Alojamiento en el hotel Puerta de Triana, la noche del miércoles al jueves Lo confirmo?
- Viaje organizado. Tren y hotel confirmados. De los restaurantes del almuerzo y la cena se ocupa la Sra. Carmen.

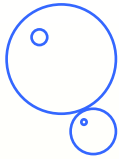
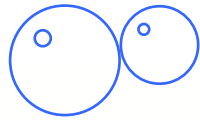




Ejemplo de funcionamiento de SMA

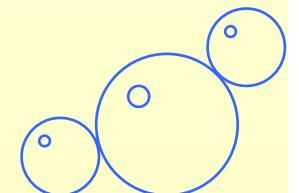
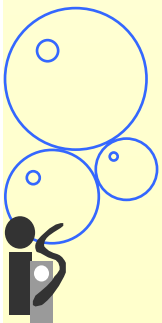
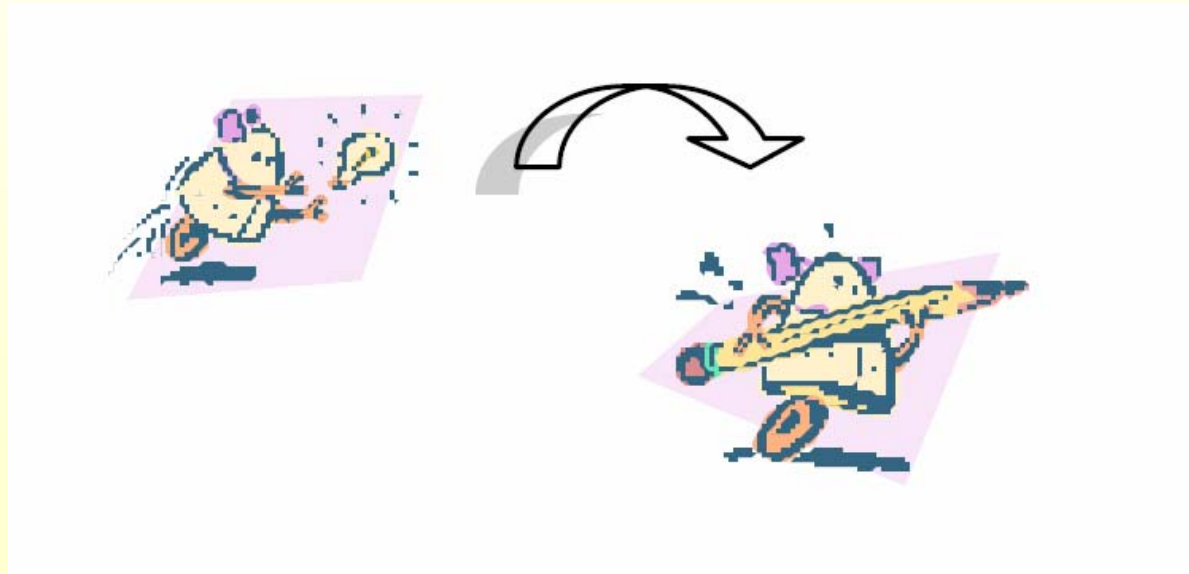
- El agente monitoriza la actividad del usuario
 - Lee/escucha la conversación del usuario
 - Reconoce patrones en la conversación
 - Deduce información y objetivos en función de experiencia pasada

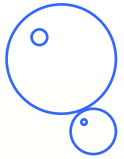
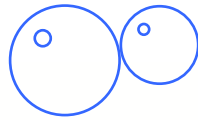




Ejemplo de funcionamiento de SMA

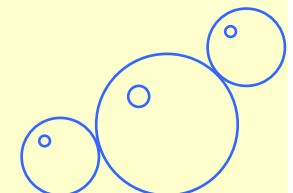
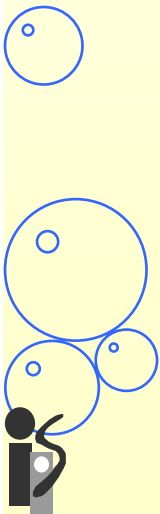
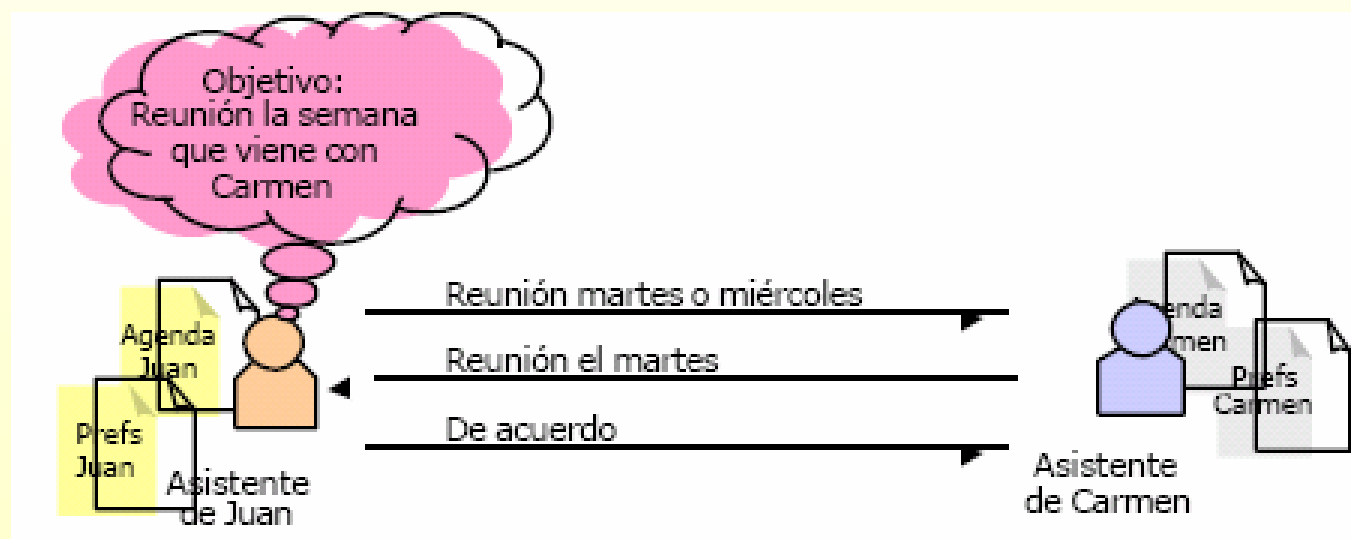
- El agente persigue satisfacer sus objetivos
 - Toma decisiones
 - Puede descomponer objetivos en subobjetivos
 - Ejecuta tareas

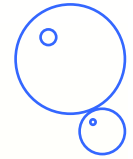
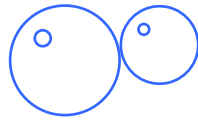




Ejemplo de funcionamiento de SMA

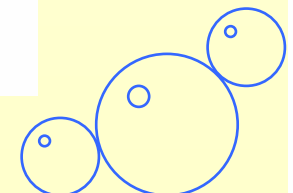
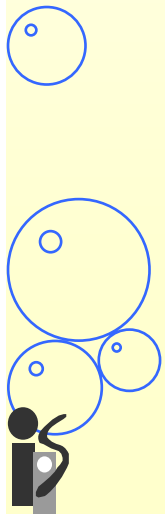
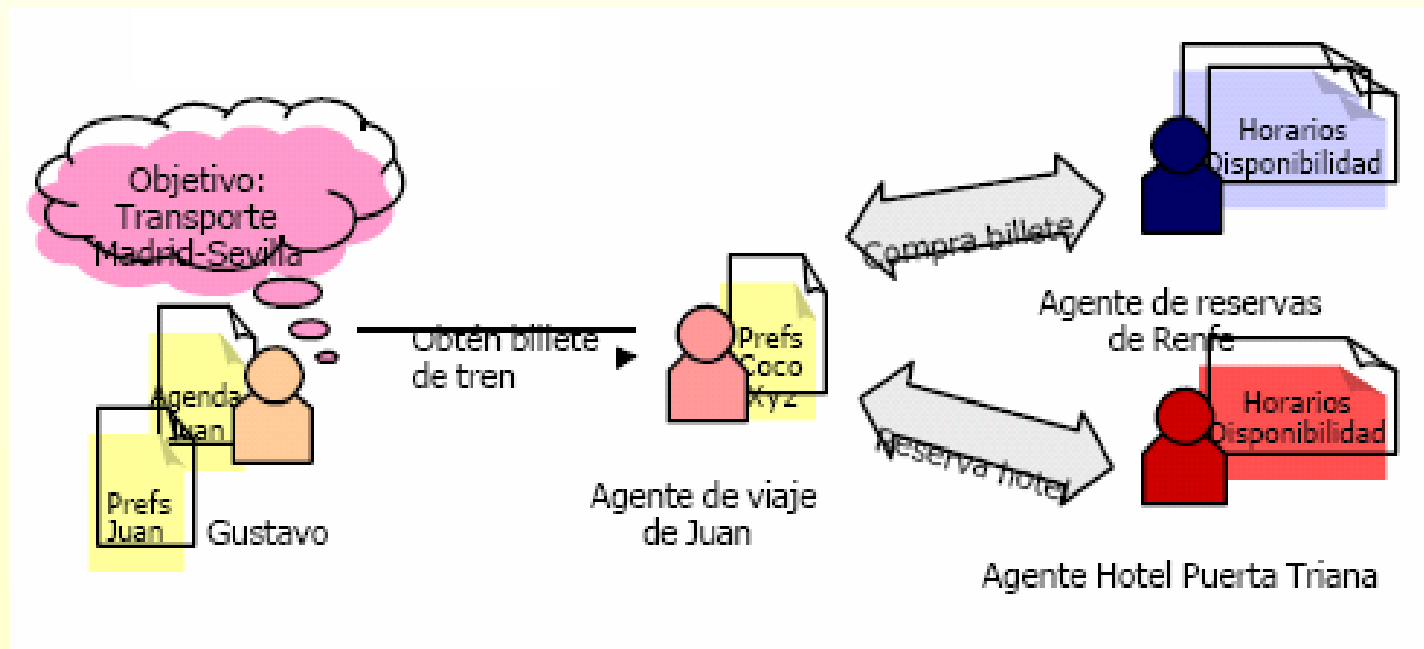
- Para cumplir objetivos necesita la colaboración con otros agentes
 - Negociación
 - Delegación
 - Coordinación

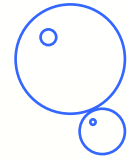
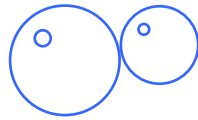




Ejemplo de funcionamiento de SMA

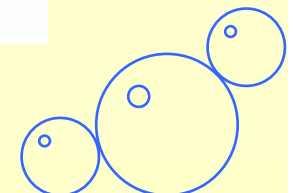
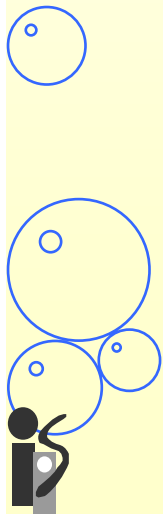
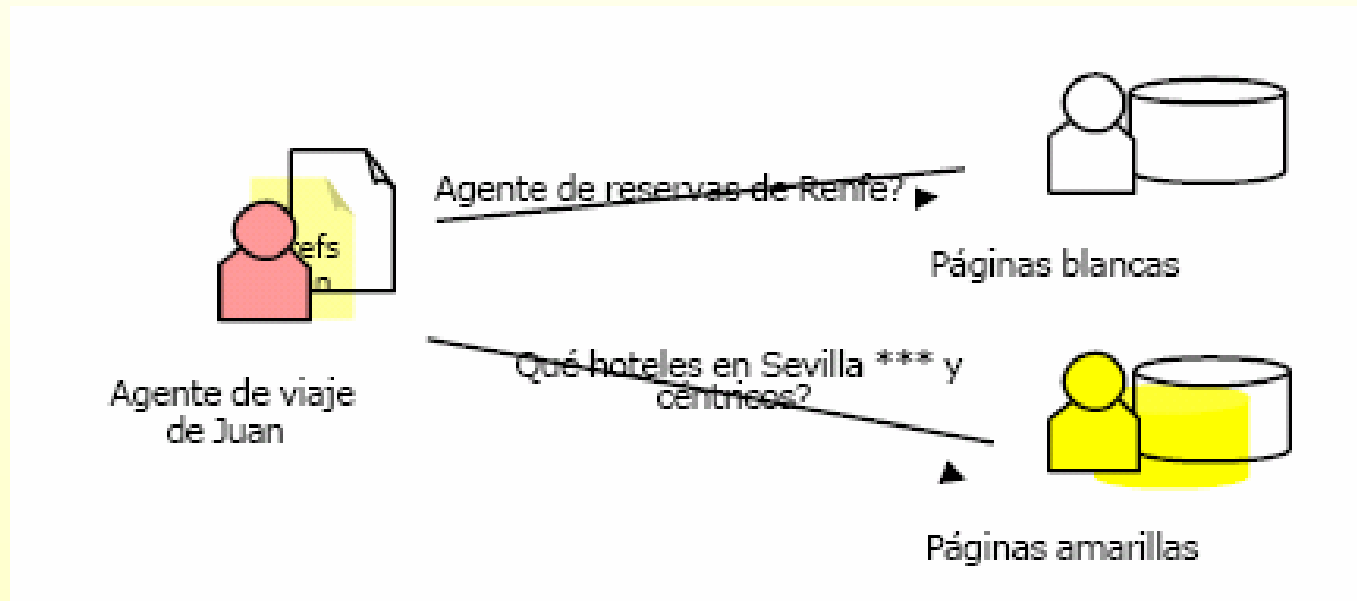
- Para cumplir objetivos necesita la colaboración con otros agentes
 - Negociación
 - Delegación
 - Coordinación

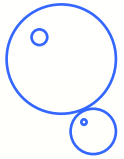
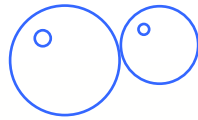




Ejemplo de funcionamiento de SMA

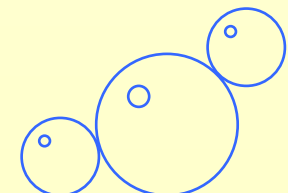
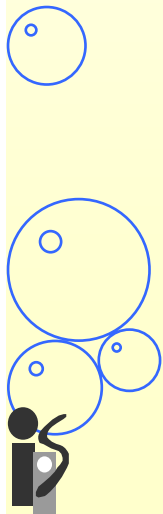
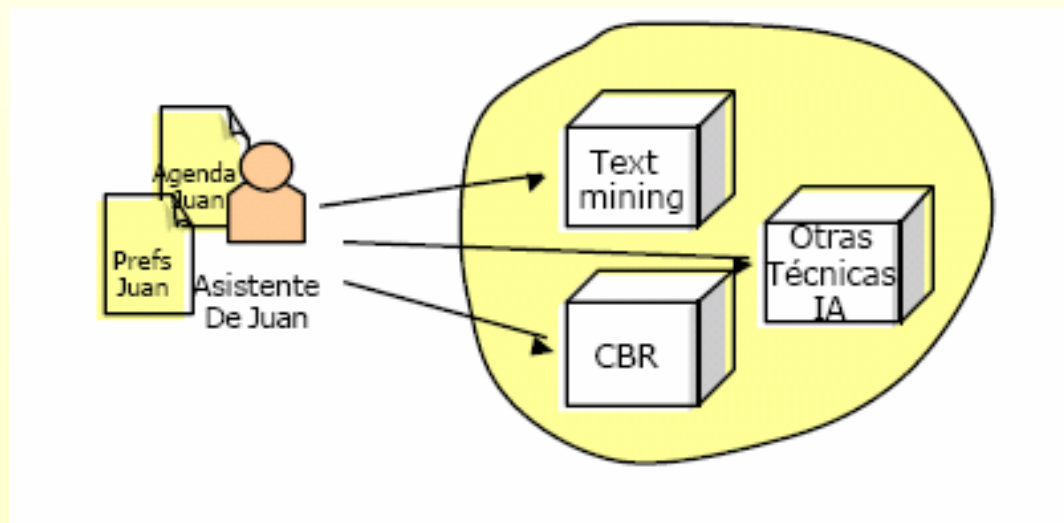
- Los agentes necesitan servicios de localización de agentes
 - Páginas blancas
 - Páginas amarillas

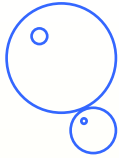
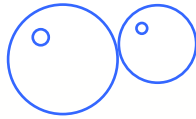




Ejemplo de funcionamiento de SMA

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

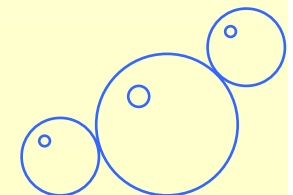
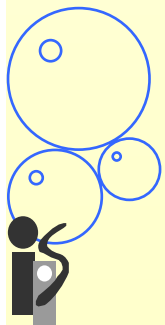


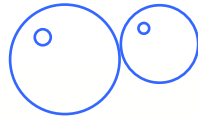


Utilidad de los sistemas multiagentes

- En el diseño de sistemas distribuidos los agentes proporcionan:
 - Aspectos sociales
 - Lenguajes y protocolos de comunicación de agentes
 - Distribución de datos, control, conocimiento, recursos
- En el análisis de un sistema los agentes tienen un mayor grado de abstracción que los objetos o componentes (Al igual de los servicios Web, pero son autónomos):
 - Mayor autonomía y capacidad de decisión
 - Varios componentes heterogéneos que mantienen relaciones entre ellos y con escalas de tiempo diferentes
 - Modelado de sistemas naturales y sociales
- Facilitan la evolución (p.e. Aplicación Grid Computing):
 - Adaptación a modificaciones y al entorno
 - Escalabilidad: añadir agentes para soportar mayor carga de trabajo
 - Añadir/quitar funcionalidad en tiempo de ejecución
 - Desarrollo incremental
 - Sistemas abiertos : capacidad de aceptar nuevos elementos

*Pero no siempre son la solución ideal
Ausencia de control/visión global del sistema*





Aplicaciones

Servicios de información en Internet
Recuperación y extracción de información

Comercio electrónico

Mercado de servicios electrónico

Negociación

Equipos móviles y PCs en el hogar

Redes públicas de telecomunicaciones

Provisión de servicios bajo demanda

Descentralización del control y gestión de redes

Grid Computing

Gestión de procesos (workflow)

Simulación de sistemas dinámicos

Juegos (bots)

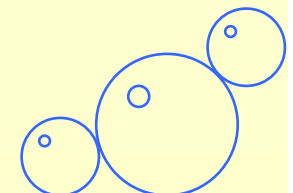
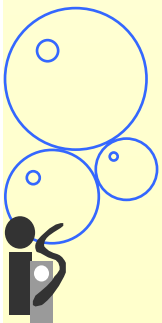
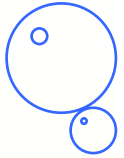
Robótica

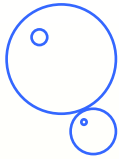
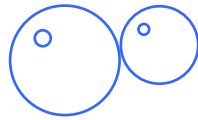
Etc.

Personalización
de servicios

Flexibilidad de
la distribución

Delegación
de tareas

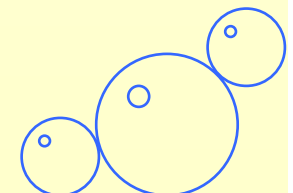
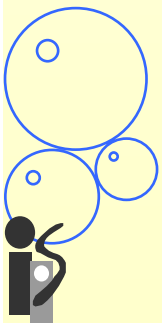


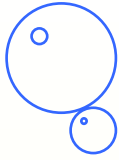
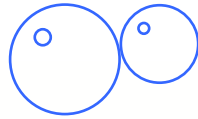


Definición SMA

Los sistemas multi-agentes que consideramos constan de:

1. Un *middleware* para soportar la comunicación entre los agentes y con sistemas propietarios
 - Sobre este middleware es posible utilizar herramientas basadas en estándares de comunicación entre agentes como FIPA ACL o KQML, o definir interfaces específicas para cada agente con un lenguaje como OMG IDL, o WSDL.
2. Agentes, de diversos tipos, que cooperan para proporcionar servicios inteligentes a los usuarios.
3. Recursos que pueden estar gestionados por agentes o por algún sistema propietario
 - Puede tratarse de un servidor Web, una base de datos, un servidor de correo, un sistema de gestión de procesos de negocio, dependiendo del entorno de la aplicación.





Objetos vs Agentes

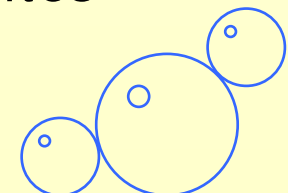
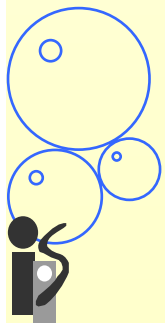
Objetos

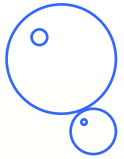
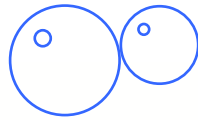
- Ejecuta los métodos invocados
- Flujo de control del llamante
- Encapsula estado y comportamiento
- Estado: valor de variables
- Comportamiento: salida a partir de una entrada
- Mensajes invocan procedimiento
- Asociaciones entre objetos

Agentes

- Autonomía de decisión
- Flujo de control propio
- Encapsula la activación del comportamiento
- Estado mental: objetivos, creencias, ...
- Comportamiento: cómo decidir lo que hacer
- Interacciones: actos de habla (intencionalidad)
- Organización: relaciones sociales entre agentes

RMR/011/urq/wd#m#g#lxvdll





Sistemas Expertos vs Agentes

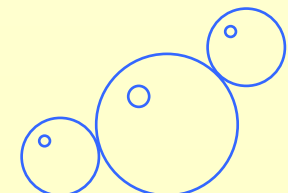
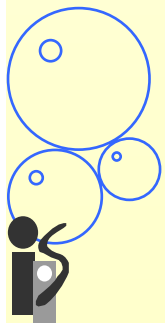
Sistemas Expertos

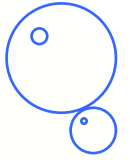
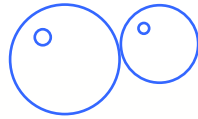
- Sistemas cerrados
- Sistemas de decisión centralizados
- Interacción con el usuario bajo petición del usuario

Agentes

- Interactúan con el entorno
- Distribución de la toma de decisiones:
 - ***Comportamiento emergente***
- Mayor grado de interacción con el usuario
- Interacción con otros agentes

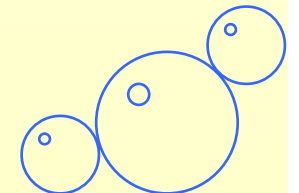
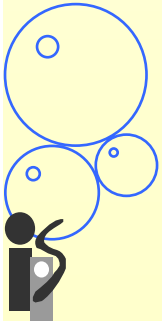
RMR #0#urqwhd#mglixvdl

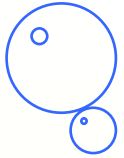
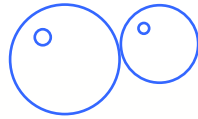




Comunicación entre agentes

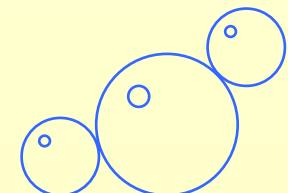
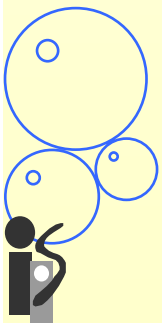
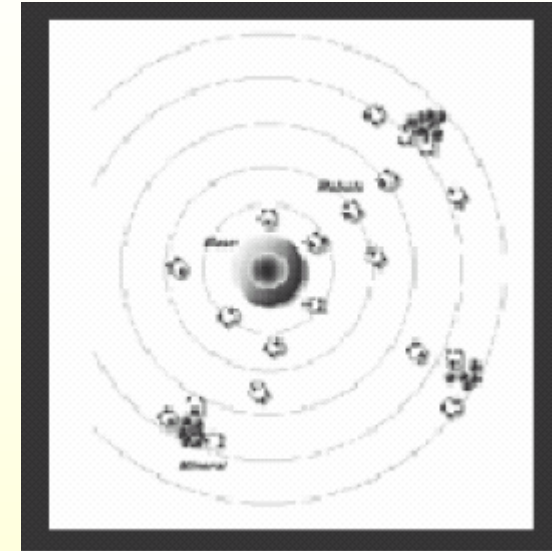
- La comunicación es la base para las interacciones y la organización social de los agentes
- Interacciones
 - **Hay interacciones cuando la dinámica de un agente está perturbada por las influencias de otros [O. Boissier, 2001]**
 - **Las interacciones son el motor de los SMA**
- Distintas formas de interaccionar
 - **Acciones sobre el entorno**
 - **Pizarra compartida**
 - **Inferencias**
 - **Paso de mensajes**
- ...

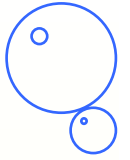
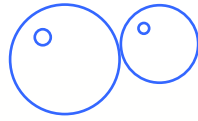




Comunicación mediante el entorno

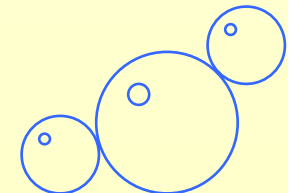
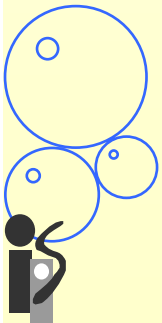
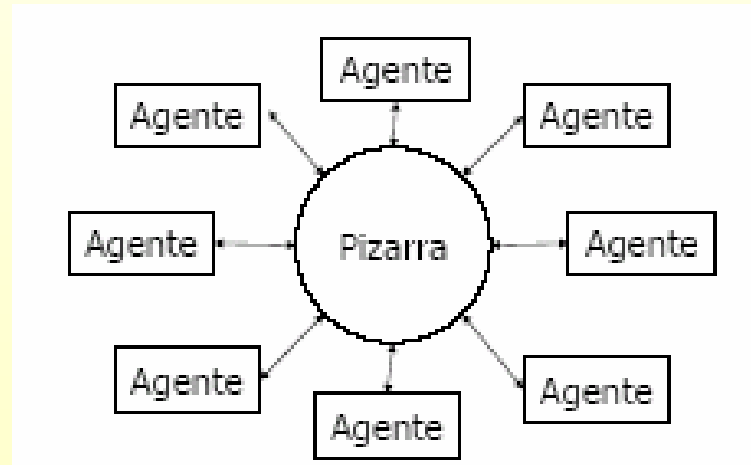
- Ejemplo: robots distribuidos [Steels 89]
Problema
Un conjunto de robots tienen que recoger piedras preciosas (cuya localización no se conoce de antemano) y llevarlas a una nave nodriza
- La comunicación se realiza a través del entorno:
 - **Campo gradiente de la señal generada por la nave nodriza**
 - **Partículas radioemisoras que pueden recoger, echar y detectar los robots al pasar**

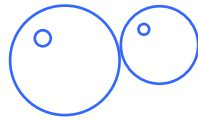




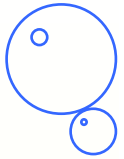
Sistema de pizarra

- Pizarra: zona de trabajo común que permite a los agentes compartir todo tipo de información
- Un sistema multiagente puede tener varias pizarras con distintos agentes registrados en cada una
- No hay comunicación directa entre agentes

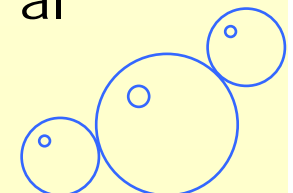
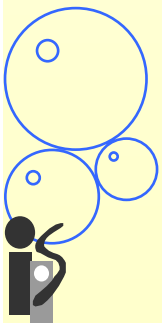


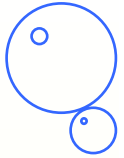
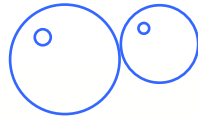


Sistema de pizarra



- Los sistemas más avanzados incorporan nuevos conceptos:
 - ***Moderador.*** Agente especializado con conocimiento de control y de evaluación que publica en la pizarra los subproblemas a resolver y decide a qué agentes se asignan de entre aquellos que se han ofrecido a resolverlos
 - ***Despachador.*** Agente que avisa a los agentes afectados por algún cambio producido en la pizarra
- Método flexible de comunicación para la resolución distribuida de problemas
 - **Son independientes de la estrategia de cooperación que se vaya a utilizar y no afectan a la arquitectura de los agentes individuales**
- Sin embargo, la estructura central de la pizarra representa cada vez más un inconveniente ya que todos los agentes distribuidos por una red se ven obligados a acceder al dispositivo central donde se encuentra la pizarra





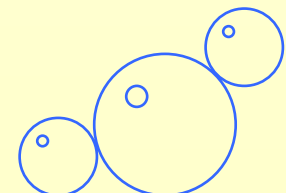
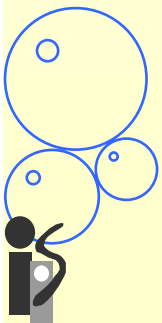
Interacción sin comunicación

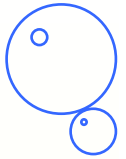
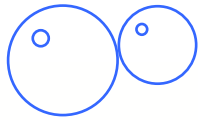
- Inferencia de acciones de otros agentes
 - Ejemplo: utilización de la teoría de juegos con matrices de ganancia [

Dilema del prisionero

Dos hombres son acusados de un crimen y encerrados en celdas separadas. No pueden comunicarse ni llegar a acuerdos. A los dos hombres se les dice

- Si uno de los dos confiesa el crimen y el otro no, el confesor será liberado, y el otro condenado a tres años.
- Si los dos confiesan, ambos serán encarcelados durante dos años.
- Si ninguno confiesa, ambos serán encarcelados durante un año.





Interacción sin comunicación

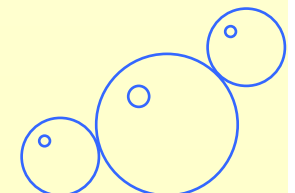
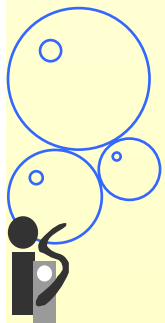
- ¿cuál es la mejor estrategia?
 - Teorías de juegos...

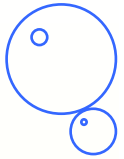
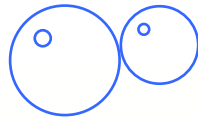
i

		Unfkd}r#	frrs
j	Unfkd}r#	5	3
	frrs	8	6

Frqv}ghud}rv#}r#}frq}hvd}#
 frp}r#}frrshud}##}frq}hvd}#
 frp}r#}hfkd}duo}#}frrshud}frq}l

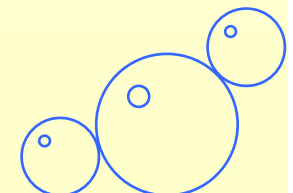
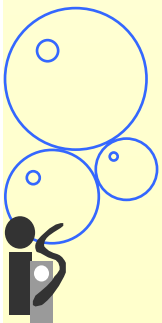
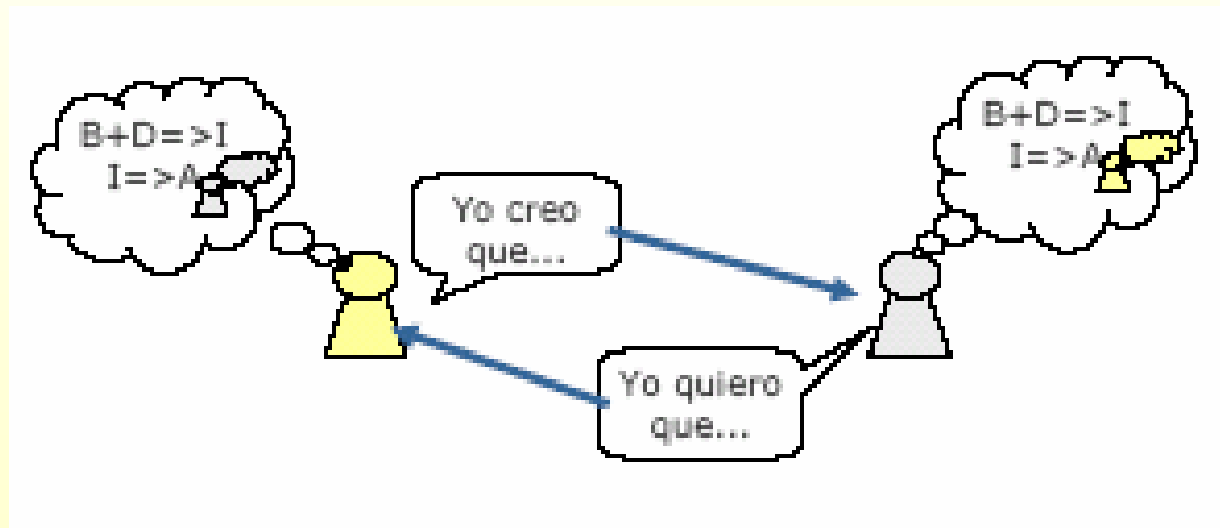
Orv#}p}hur}v#}r#}bg}fd}e}v#}d}..rv#
 v}r#}e}x}h}r#}x}#}v
 sd}e}v#}d}j}h}w}v

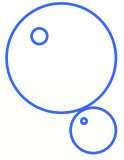
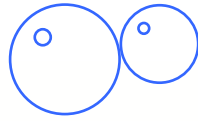




Comunicación a nivel del conocimiento

- Los agentes se comunican para
 - Mostrar a otros agentes su estado mental
 - Intentar modificar el estado mental de otros agentes



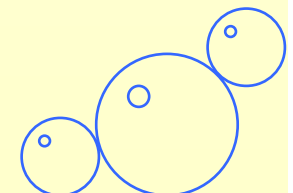
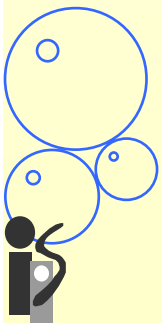


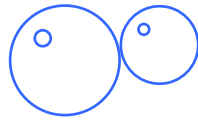
Actos del habla (Speech acts)

- Language as Action
 - **J.L. Austin (1962), How to do things with words, Clarendon Press**

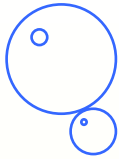
La lingüística tradicional intentaba entender el significado de las frases indicando cómo es posible usar una combinación de palabras para hacer una declaración con significado

- **Los actos del habla hacen referencia a las acciones intencionales en el curso de una conversación**





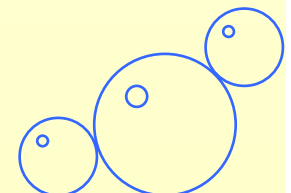
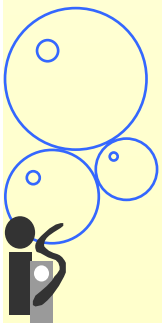
Actos del habla (Speech acts)

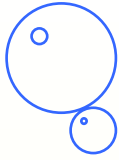
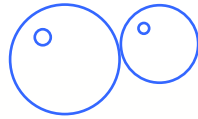


Language as Action

– **J.L. Austin (1962), How to do things with words, Clarendon Press**

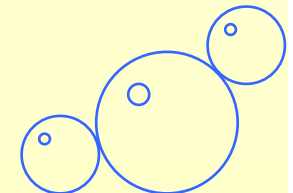
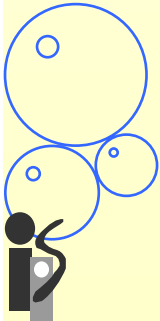
- Quien habla no declara solamente sentencias ciertas o falsas
- Quien habla realiza actos de habla:
peticiones, sugerencias, promesas, amenazas, etc.
- Cada declaración es un acto de habla

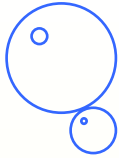
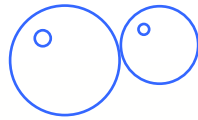




Tipos de actos del habla

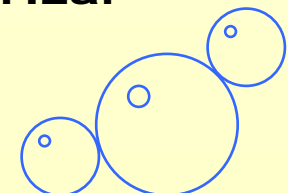
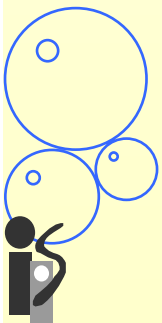
- J.R. Searle (1969), *Speech Acts*, Cambridge University Press
- Actos asertivos: dan información sobre el mundo
 - **Estoy de acuerdo** **2 y 2 son 4** **Estamos en clase**
- Actos directivos: para solicitar algo al destinatario
 - **Siéntate** **¿Cuántas pesetas son un euro?**
- Actos de promesa: comprometen al locutor a realizar ciertas acciones en el futuro
 - **Mañana vuelvo a las 8** **Te enviaré las fotos**
- Actos expresivos: dan indicaciones del estado mental del locutor
 - **Soy feliz** **Gracias** **Siento lo de tu perro**
- Actos declarativos: el mero hecho de la declaración es la realización de un acto
 - **Estás contratado** **Empezamos la clase**

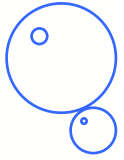
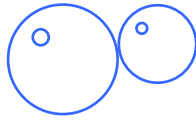




Componentes de los actos del habla

- Locución: modo de producción de frases utilizando una gramática y un léxico
- Ilocución: acto realizado por el locutor sobre el destinatario mediante la declaración (utterance)
 - **Fuerza ilocutoria (F): afirmación, pregunta, petición, promesa, orden => PERFORMATIVA**
 - **Contenido proposicional (P), objeto de la fuerza ilocutoria**
 - **Se puede representar como F(P) performativa(contenido))**
aserta(está nevando) responde(está nevando)
- Perlocución: efectos que pueden tener los actos ilocutorios en el estado del destinatario y en sus acciones, creencias y juicios
Ejemplos: convencer, inspirar, persuadir, atemorizar



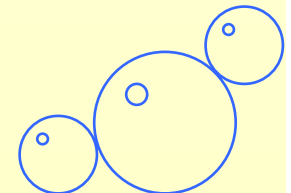
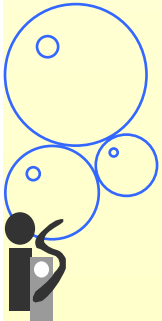


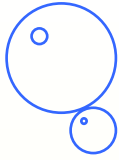
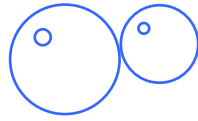
Componentes de los actos del habla

- Ejemplo:

Cierra la puerta

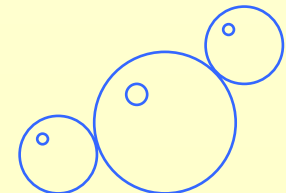
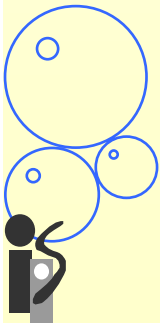
- locución: declaración física con contexto y referencia: quién habla y quién escucha, qué puerta, etc.
- ilocución: acto de llevar intenciones: el que habla quiere que el que escucha cierre la puerta
- perlocución: acciones que ocurren como resultado de la ilocución: el que escucha cierra la puerta

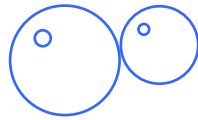




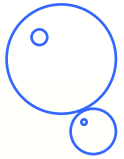
Éxitos de los actos del habla

- Una declaración no es verdadera o falsa: tiene éxito o fracasa
- Un acto de habla puede fallar
 - en su enunciación, porque no llegue el mensaje o llegue corrompido o el destinatario no lo entiende
 - en su interpretación, por el destinatario
 - en su consecución final, por ejemplo porque el destinatario no sea capaz de llevar a cabo la acción solicitada

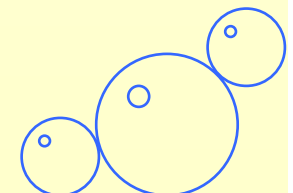
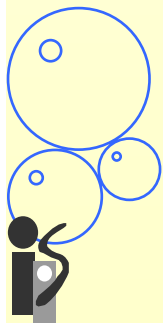
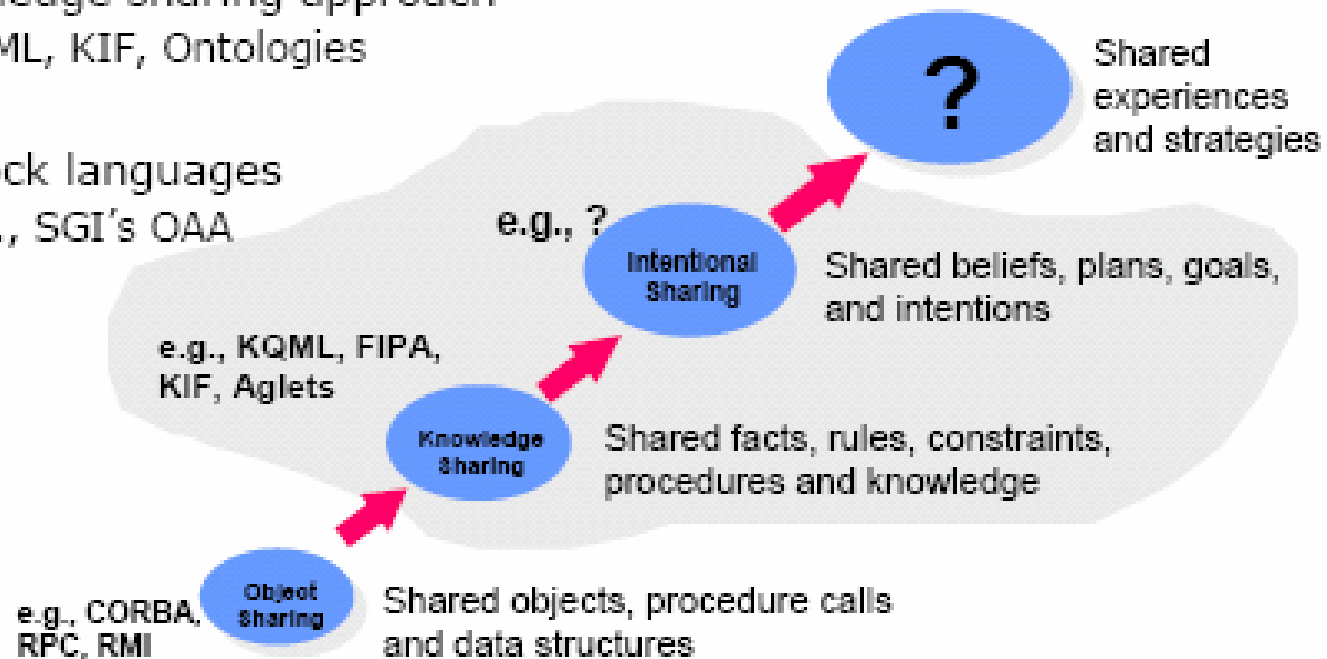


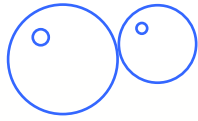


Lenguajes de comunicación de agentes (Y. Labrou 2000)



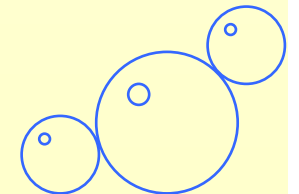
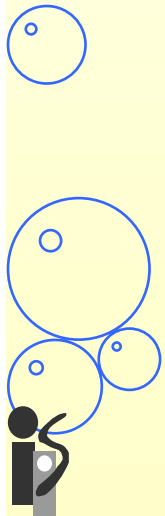
- Es CORBA un ACL?
- Knowledge sharing approach
 - KQML, KIF, Ontologies
- FIPA
- Ad hock languages
 - e.g., SGI's OAA

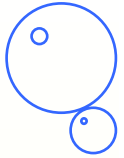
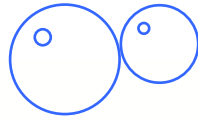




Knowledge Sharing Effort

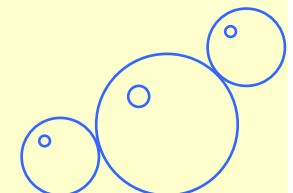
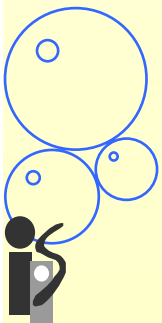
NMH

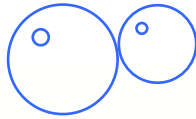




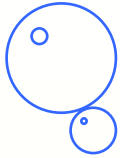
KSE

- Iniciado por ARPA hacia 1990, y apoyado por organismos norteamericanos de investigación (ASOFR, NSF, NRI)
- Propósito:
 - Desarrollo de técnicas, metodologías y herramientas software para la compartición y reutilización del conocimiento entre sistemas
- **El KSE produjo dos especificaciones**
 - KQML (knowledge Query Manipulation Language): Define la envoltura de un mensaje. El agente muestra explícitamente su intención (illocución).
 - KIF (Knowledge Interchange Format): Representación del conocimiento. Contenido del mensaje.

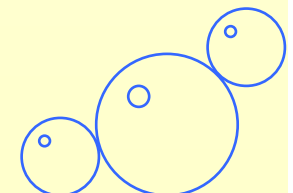
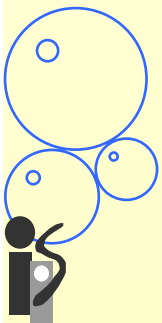


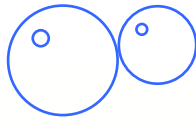


KSE



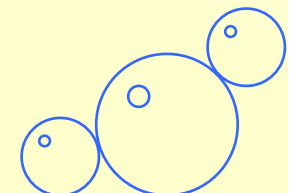
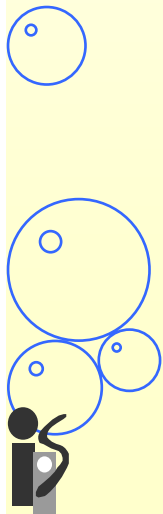
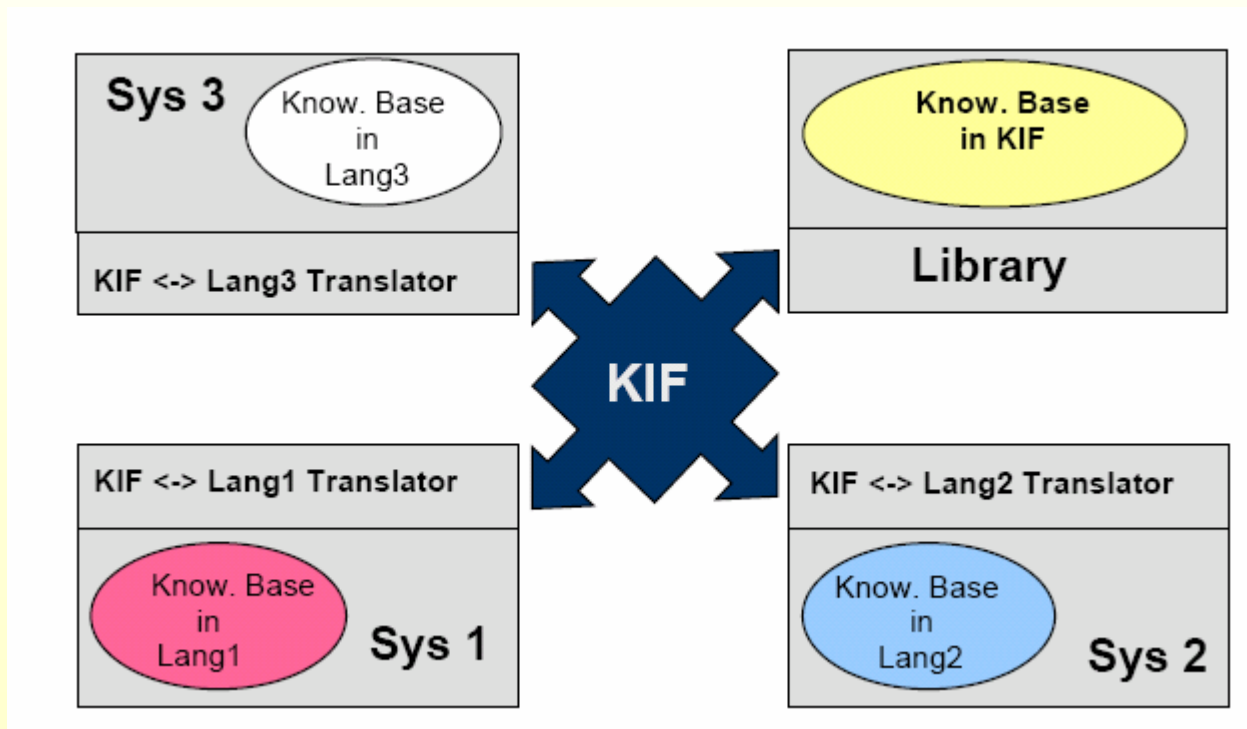
- Compartir conocimiento entre agentes requiere la capacidad de comunicarse. La capacidad de comunicarse requiere un **lenguaje de comunicación común**:
 - Sintaxis: KIF (Knowledge Interchange Format)
 - Semántica: Ontolingua (lenguaje para definir **ontologías**)
 - Pragmática: KQML (Knowledge Query and Manipulation Language, lenguaje de interacción de alto nivel)

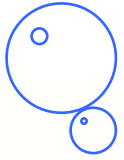
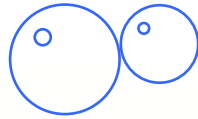




KIF

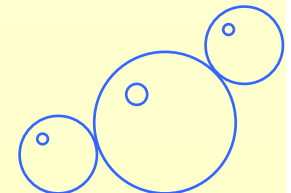
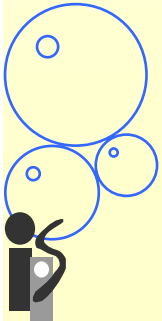
- Objetivo: compartir bases de conocimiento mediante un lenguaje común (interlingua)





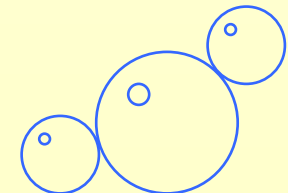
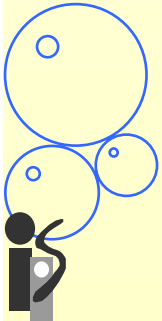
KIF

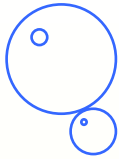
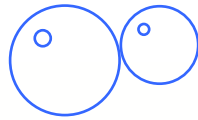
- Aspectos sintácticos de la representación del conocimiento para expresar
 - **Conocimiento**
 - **Meta-conocimiento**
- basado en cálculo de predicados de primer orden y teoría de conjuntos, con extensiones para soportar:
 - **razonamiento**
 - **definiciones**
- Es independiente de la implementación y legible por humanos
- Especificación actual en <http://logic.stanford.edu>



KIF

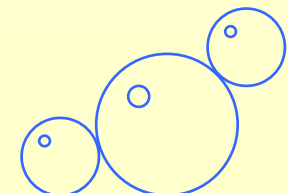
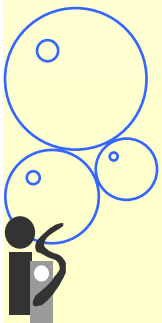
- Ejemplos de expresiones
 - **Datos**
 - (empleado pepe ventas)
 - tupla que indica que pepe es un empleado en el departamento de ventas
 - **Operaciones lógicas**
 - (> (* (ancho chip1) (largo chip1)) (* (ancho chip2) (largo chip2)))
 - establece que un chip es de mayor tamaño que el otro
 - **Representación conocimiento**
 - (interesado pepe '(empleado ,?x ,?y))
 - el agente pepe está interesado en recibir tuplas de la relación de empleados
 - **Procedimientos (con sintaxis parecida a LISP)**
 - (progn (frase-tipica f)
 - (print "Hola mundo.")
 - (frase-tipica f))

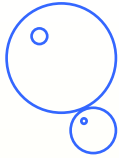
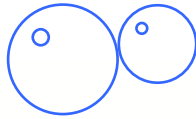




Ontololingua

- Ontolingua es un lenguaje que permite construir, publicar y compartir ontologías
 - **Una interfaz a un servidor de edición/navegación**
 - **Las ontologías pueden traducirse automáticamente a distintos lenguajes de contenidos, como KIF, LOOM, Prolog, CLIPS, etc.**
- El lenguaje incluye primitivas para combinar ontologías
- Otras herramientas para trabajar con ontologías:
 - **Ontosaurus (<http://www.isi.edu/isd/ontosaurus.html>), navegador para Loom**
 - **WebOnto (<http://kmi.open.ac.uk/projects/webonto/>), Java applet**





Convergencia de intereses con los Servicios Web

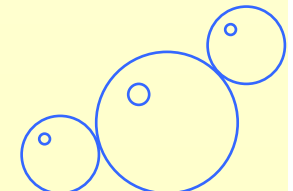
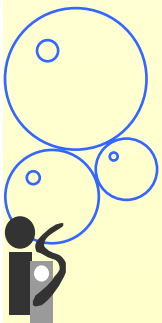
- Web Semántica
Weners-Lee, el padre del World-Wide Web (HTTP+HTML), ha propuesto su siguiente visión: **La Web Semántica**

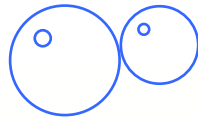
La Web :

- Compartir entre humanos conocimiento en forma de documentos hiperenlazados
- La interpretación es posible a que los humanos compartimos conocimientos.

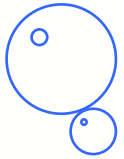
La Web Semántica:

- Compartir entre aplicaciones conocimientos. Se trata de etiquetar las páginas no sólo con detalles de presentación (HTML), sino con detalles de significado.

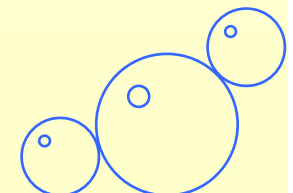
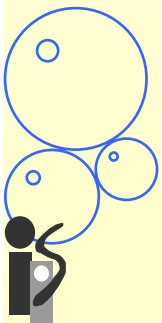


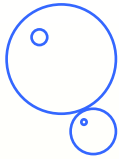
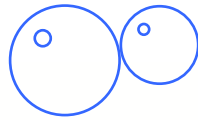


Ontologías



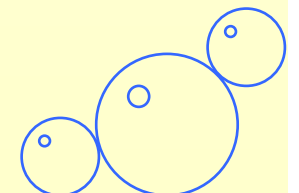
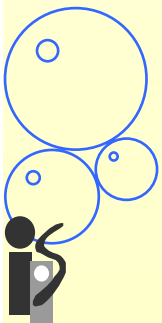
- Ontología: un vocabulario común en el que se han acordado significados para describir un dominio
 - **Es una conceptualización del mundo, en función de objetos, cualidades, distinciones, relaciones**
- Una ontología define un conjunto de clases, funciones y constantes para un dominio de discurso, e incluye una axiomatización para restringir su interpretación (reglas de inferencia)
- Ejemplos de ontologías:
 - **Cyc (<http://www.cyc.com/>)**, ontología de propósito general
 - **WordNet**, sistema de referencia léxica accesible por internet
 - **CIA World Fact Book (<http://www.cia.gov/cia/publications/factbook/>)**, 5 megas de KIF... Servicios Web
 - **UMLS (Unified Medical Language System)**

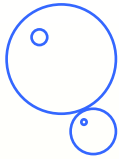
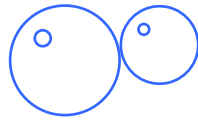




Ontologías: Una definición

- **Una ontología es un modelo computacional de alguna porción del mundo.**
 - Se captura como red semántica: Un grafo con conceptos o objetos individuales, cuyos arcos representan relaciones o asociaciones de objetos.
 - La red se complementa con propiedades y atributos, restricciones, funciones, y reglas que gobiernan el comportamiento de los conceptos.
- **Formalmente una ontología es un acuerdo sobre una conceptualización compartida para modelar el dominio de conocimiento y sobre la representación.**

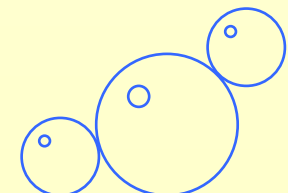
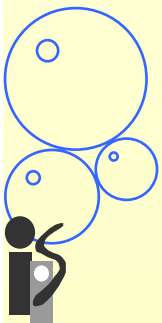


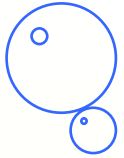
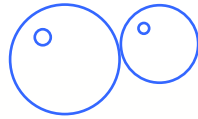


Representaciones del conocimiento

- **Esquemas de clasificación de conceptos**

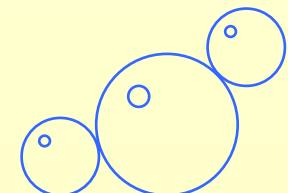
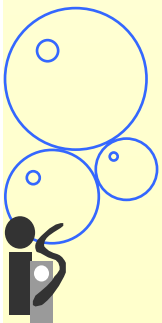
- **Keywords:** Son la orma más rápida y sencilla de localizar información potencialmente útil. Metadatos básicos.
- **Tesauros:** Ofrecen una aproximación estructurada al manejo de las palabras clave, estableciendo entre los conceptos las relaciones broader, narrower, and related.
- **Taxonomías:** Estructuras de clasificación que añaden la potencia de la herencia.
- **Ontologías:** Mayor riqueza en las relaciones.
 - La mayoría de ontologías incoporan las relaciones:
 - Generalización y Herencia
 - Agregación.
 - Instanciación

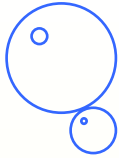
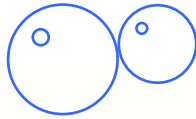




Frames vs Descriptions

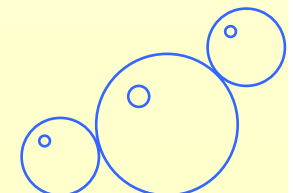
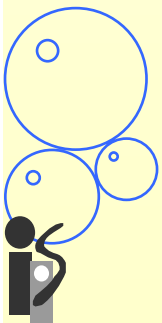
- **Dos aproximaciones detrás de las ontologías:**
 - **Frames:** Representación del conocimiento directamente en terminos de grafos. Se deninen los frames (que representan clases o instancias) y se relacionan entre ellos. Los frames se definen directamente en terminos de un conjunto de propiedades (slots) y propiedades (facets).
 - **Description Logics** es una familia de lenguajes que formalmente expresan ciertas restricciones sobre la representación del conocimiento. Tienen una semántica precisa y se puede automatizar su procesamiento. Description logics comienza con conceptos primitivos y sigue con la definición de conceptos en términos de descripciones formales. La **Subsumption**, esto es la jerarquía de especialización entre conceptos, se puede determinar de la descripción de los conceptos.
 - Description logic determina si un concepto es redundante o está relacionado con otros conceptos.

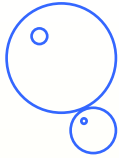
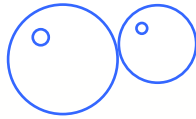




RDF (Resource Description Framework)

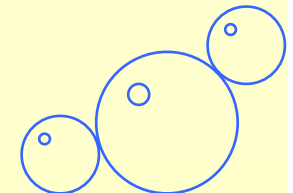
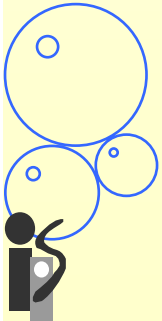
- **RDF es un lenguaje que permite en XML definir estructuras tipo grafo. De esta forma cualquier conocimiento estructurado se puede representar en RDF**
 - Un documento RDF es una colección de sentencias expresadas como tripletas. El conocimiento se expresa en forma de tripletas subject, predicate, object p.e . (estudiante, subclase, persona)

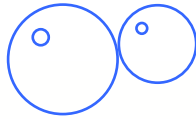




Web Ontology Language

- **Web Ontology Language (OWL) permite especificar clases y propiedades en forma de sentencias description logic.**
- **OWL-S: Una ontología para la descripción de los servicios Web.**
- **SWRL: Semantic Web Rule Language. Extiende OWL con reglas (RuleML).**

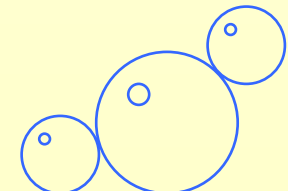
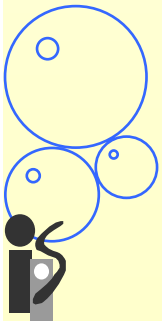


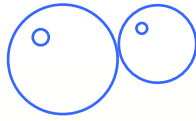


KQML (Knowledge Query and Manipulation Language)

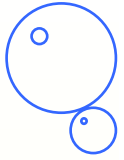


- Es un lenguaje de comunicación y protocolo, orientado a mensajes, para el intercambio de información
- KQML es independiente de
 - **protocolos de transporte (TCP/IP, HTTP, ...)**
 - **sintaxis de contexto**
 - **ontologías**
 - **protocolos de alto nivel (contract net, subasta, ...)**

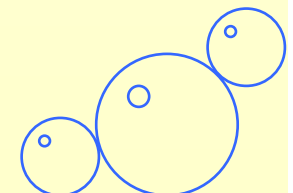
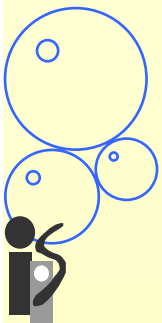




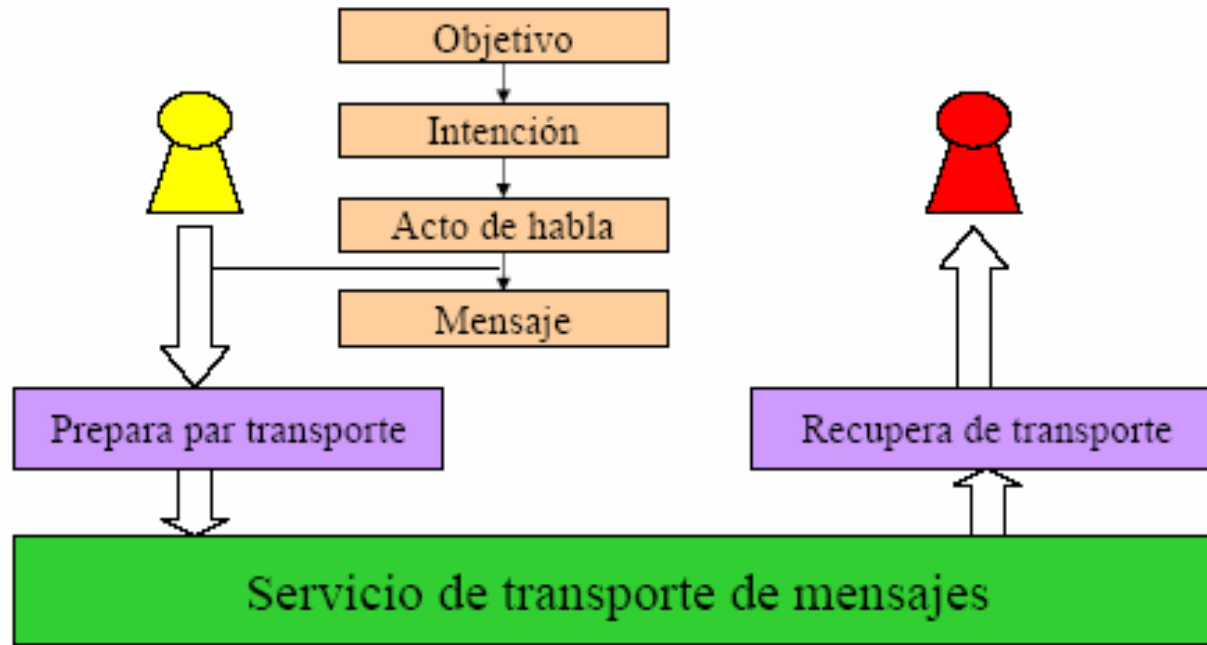
KQML

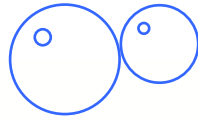


- Asume un modelo de agentes:
 - entidades de alto nivel con capacidades cognitivas (representación simbólica, base de conocimientos, ...)
 - tienen una descripción de nivel intencional: su estado es un conjunto de componentes mentales como creencias, capacidades, elecciones, compromisos, etc.
 - los agentes residen en el nivel del conocimiento
- Los mensajes KQML comunican una actitud sobre el contenido que llevan (aserto, solicitud, pregunta)
 - Las primitivas del lenguaje se llaman performativas
 - Cada mensaje KQML representa un acto de habla

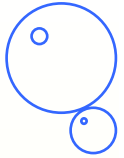


KQML

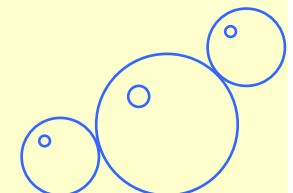
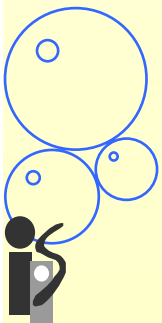




KQML



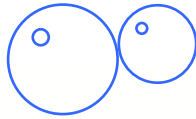
- Nivel de contenidos
 - Lleva el contenido del mensaje en el lenguaje de representación propio de los agentes
 - No es procesado por las implementaciones de KQML (se deja al agente)
- Nivel de mensajes
 - Núcleo del lenguaje KQML, determina los tipos de interacción que pueden tenerse con un agente que hable KQML
 - Identifica el protocolo que se utiliza para enviar un mensaje y proporciona una performativa al emisor que añade el contenido
 - Indica también el lenguaje de contenidos, ontología asumida, y algún descriptor sobre el contenido (esto permite a algunas aplicaciones procesar los mensajes sin acceder al contenido)
- Nivel de comunicaciones
 - Trata los parámetros de comunicación de más bajo nivel: identidad de emisor y receptor, identificación de la comunicación



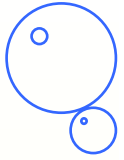
KQML

- Mensajes KQML
 - Representa un acto de habla o performativa
 - Consta de una lista de pares atributo-valor





KQML



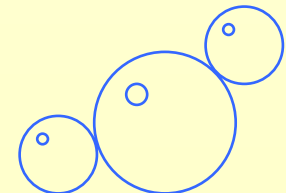
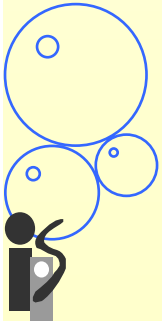
- Mensajes KQML

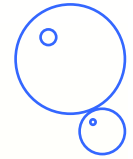
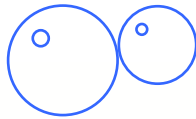
La respuesta al anterior:

(tell

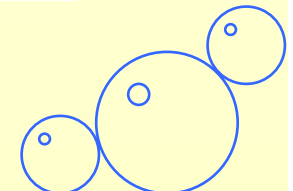
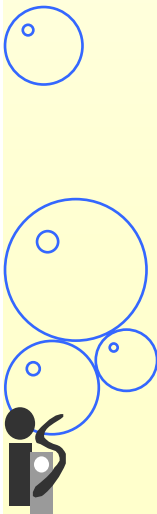
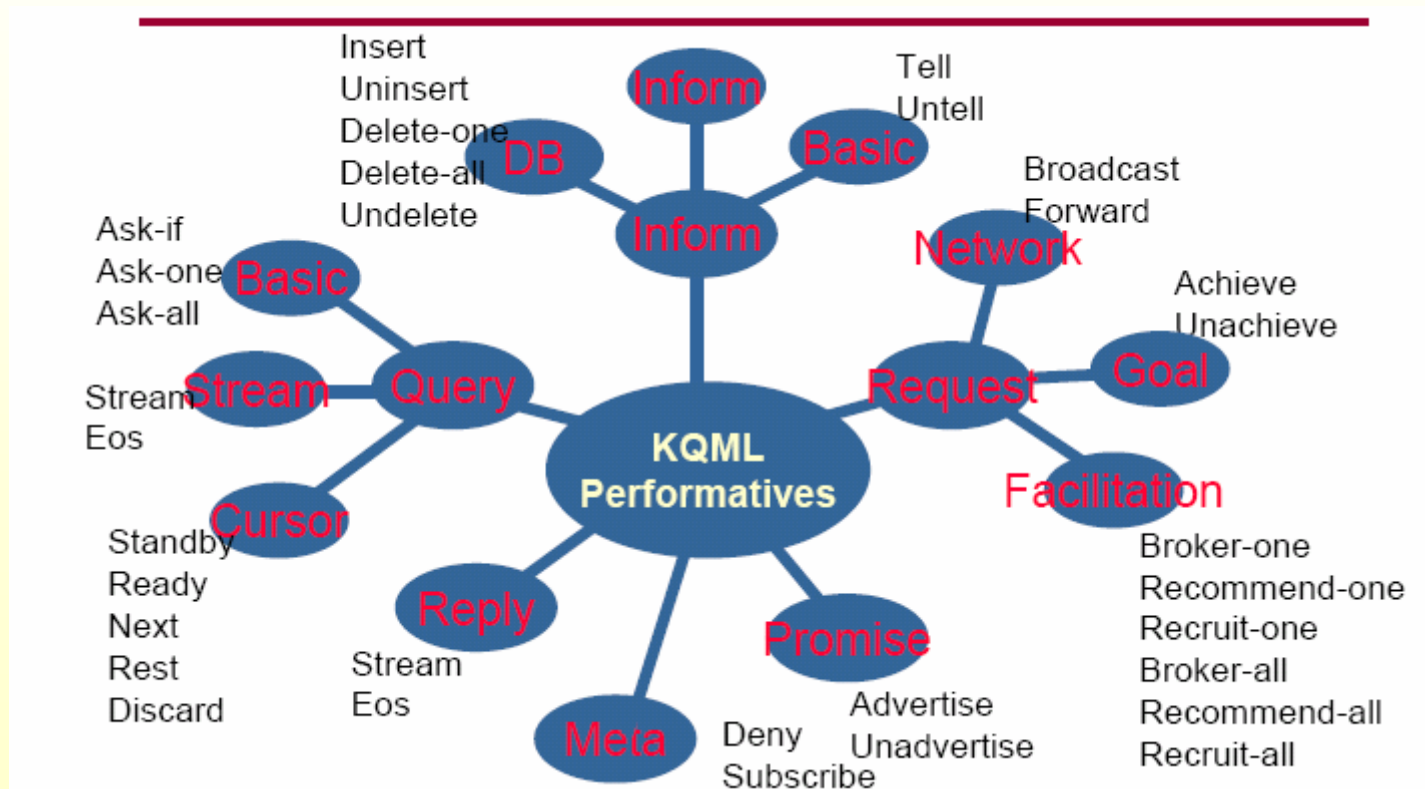
:sender servidor-bolsa
:content (PRECIO TELEFONICA 19)
:receiver pepe
:in-reply-to accion-telefonica
:language LPROLOG
:ontology IBEX

)





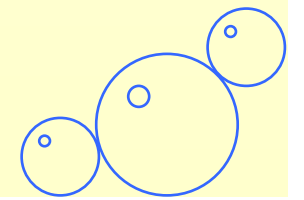
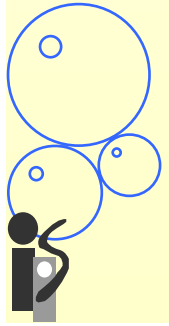
Performativas de KQML

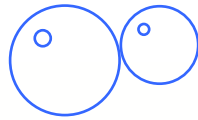


FIPA

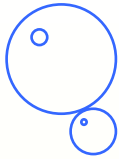


Irrogwlrq inuIqwhojhqw
Sk|vifobDjhqw +IISD,#

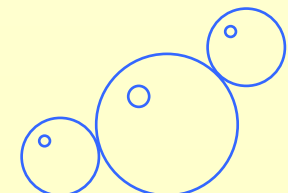
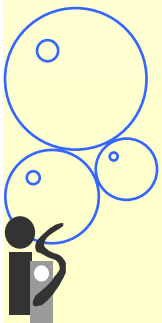


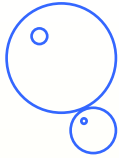
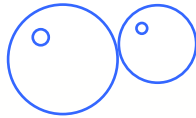


Arquitectura Abstracta FIPA



- Foundation for Intelligent Physical Agents
 - **Consortio industrial fundado en 1996**
 - **Varias decenas de compañías de telecomunicaciones e informática**
<http://www.fipa.org>
- **Objetivos**
 - **Acelerar el desarrollo de tecnologías de agentes inteligentes mediante la producción de especificaciones acordadas internacionalmente**
- **Especificación del comportamiento y capacidades externas de subsistemas genéricos: recursos de agentes (para migración, ejecución, etc.), interacción y cognitivos**
 - **las especificaciones estarán basadas en casos prácticos concretos**
 - **Agentes, multi-agentes, y sociedades de agente**
 - **Selección y adaptación de tecnologías existentes**





Arquitectura Abstracta FIPA

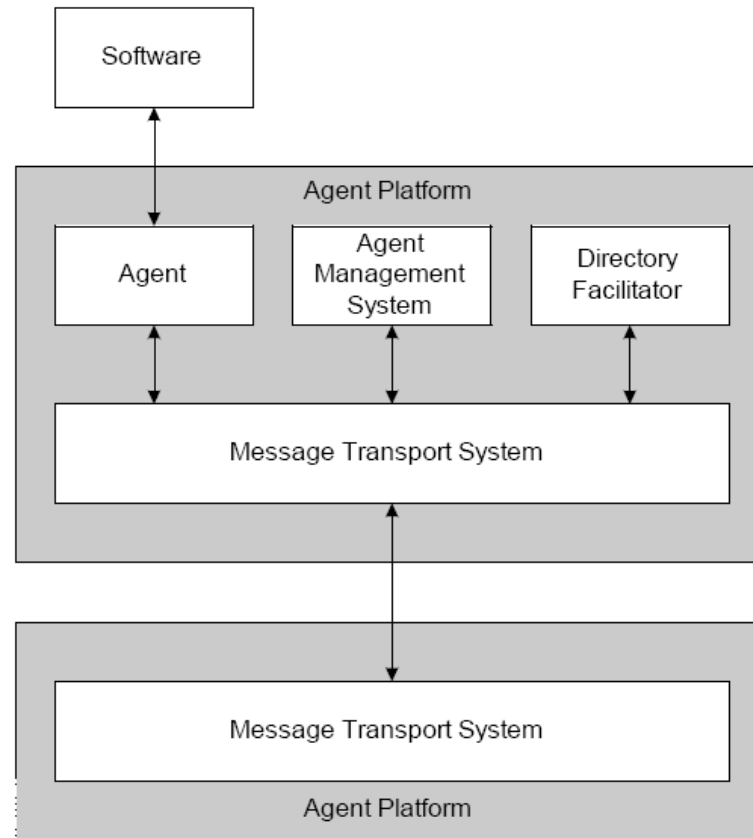
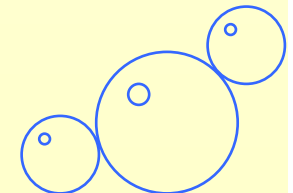
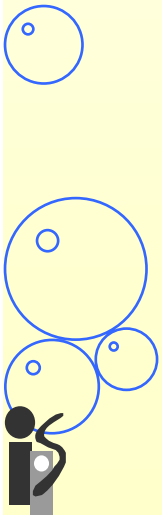
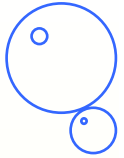
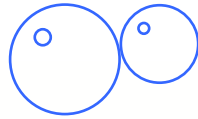


Figure 1: Agent Management Reference Model



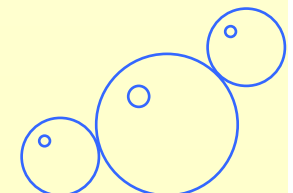
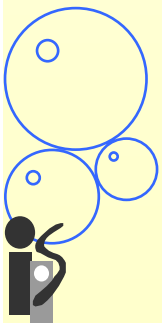


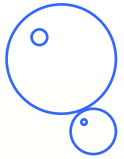
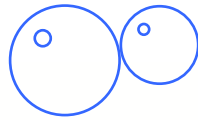
Arquitectura Abstracta FIPA

Un **Agente** (agent) es un proceso computacional que implementa la funcionalidad de comunicación y autonomía de una aplicación. Los agentes comunican utilizando el "Agent Communication Language". An El agente es el actor principal de una plataforma de agentes (AP). La AP ofrece servicios de publicación, y descripción de servicio. Un agente debe tener al menos un propietario y una noción de identidad única. La identidad viene dada por el Agent Identifier (AID). El agente puede ser registrado con distintos protocolos (transport addresses) de transporte.

El directorio (**Directory Facilitator, DF**) es una componente opcional. Ofrece servicios de páginas amarillas en las que registrar o preguntar por servicios de otros agentes. Pueden existir varios DF en una plataforma y estar federadas.

Un sistema de gestión de agentes (**Agent Management System, AMS**) es una componente obligatoria de la AP. Ejerce el control sobre el acceso y uso de la AP. Solo puede existir un AMS en la AP y mantiene un directorio de AIDs que contiene las direcciones de transporte (entre otras cosas) de los agentes registrados .

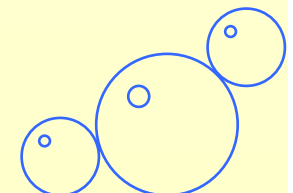
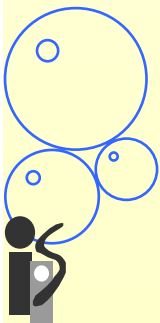


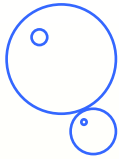
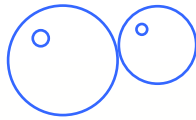


Arquitectura Abstracta FIPA

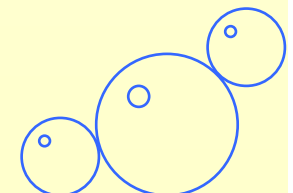
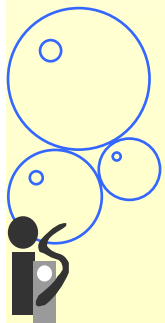
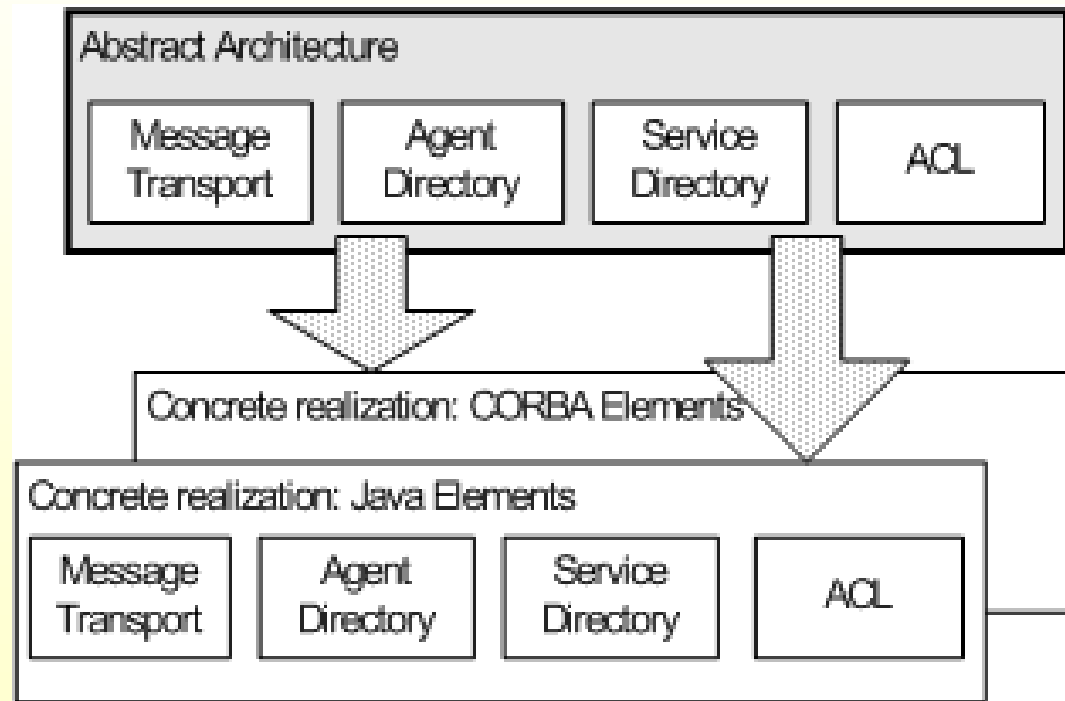
Un servicio de transporte de mensajes (**Message Transport Service , MTS**) es el método de comunicación por defecto entre los diferentes agentes.

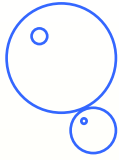
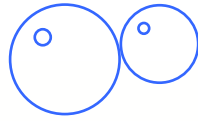
Una plataforma de agentes (**Agent Platform, AP**) suministra la infraestructura física en la que se despliegan los agentes. Consiste en sistema operativo, soporte para ejecutar los agente, y los componentes FIPA (DF, AMS, MTS).





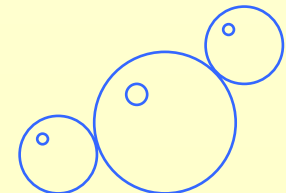
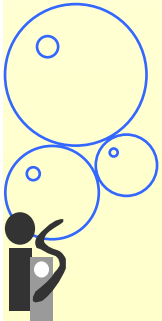
Arquitectura Abstracta FIPA

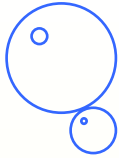
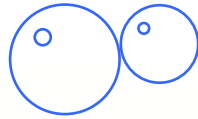




Arquitectura Abstracta FIPA

- Los agentes se comunican intercambiando mensajes
 - **que representan actos de habla**
 - **codificados en un lenguaje de comunicación de agentes**
- Servicios de soporte a los agentes:
 - **Servicios de directorio**
 - **Servicios de transporte de mensajes**
- Los servicios se pueden implementar
 - **como agentes**
 - **como software que se accede invocando métodos:**
 - C++, Java, IDL



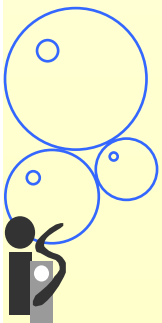


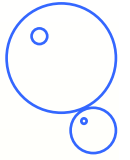
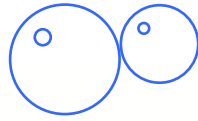
Arquitectura Abstracta FIPA

Servicios de Directorio

- Soporta un conjunto de entradas formadas por varias tuplas cada una con dos pares clave-valor:
 - **Nombre de agente:** único globalmente
 - **Localizador:** Uno o más descriptores de transporte que describen el tipo de transporte y la dirección de transporte específica para comunicarse con el agente
- Los agentes registran entradas de directorio para que otros agentes puedan buscarlas con el propósito de encontrarles para poder interactuar
 - **Ejemplo de entrada de directorio:**

```
Qrpeuh#gh#bjhqw#DjhqwHhp sor
Orfdj}dgru#          wsr#gh#wbvsruw#  glhff%o#gh#wbvsruw
                    KWS                    kws=2zzz.bjhqwv1frp 2DjhqwHhp sor
                    VPWS                   DjhqwHhp sor C djhqwv1frp
Dwlexrv#gh#bjhqw#glp dv#Hjdvk/#Ivsd.ro
Qhjrfd%o#shhugd#Erwdfw@hw
```

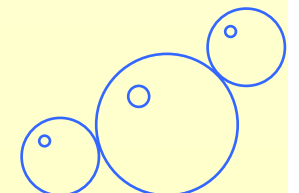
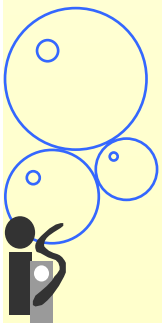


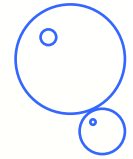
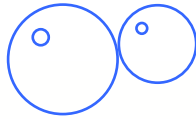


Arquitectura Abstracta FIPA

Mensajes de los agentes

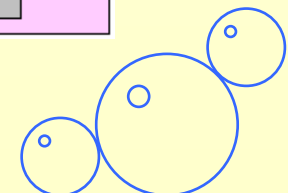
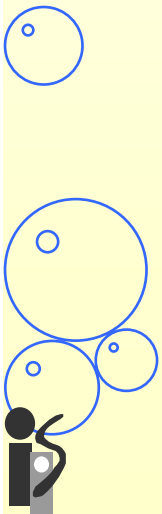
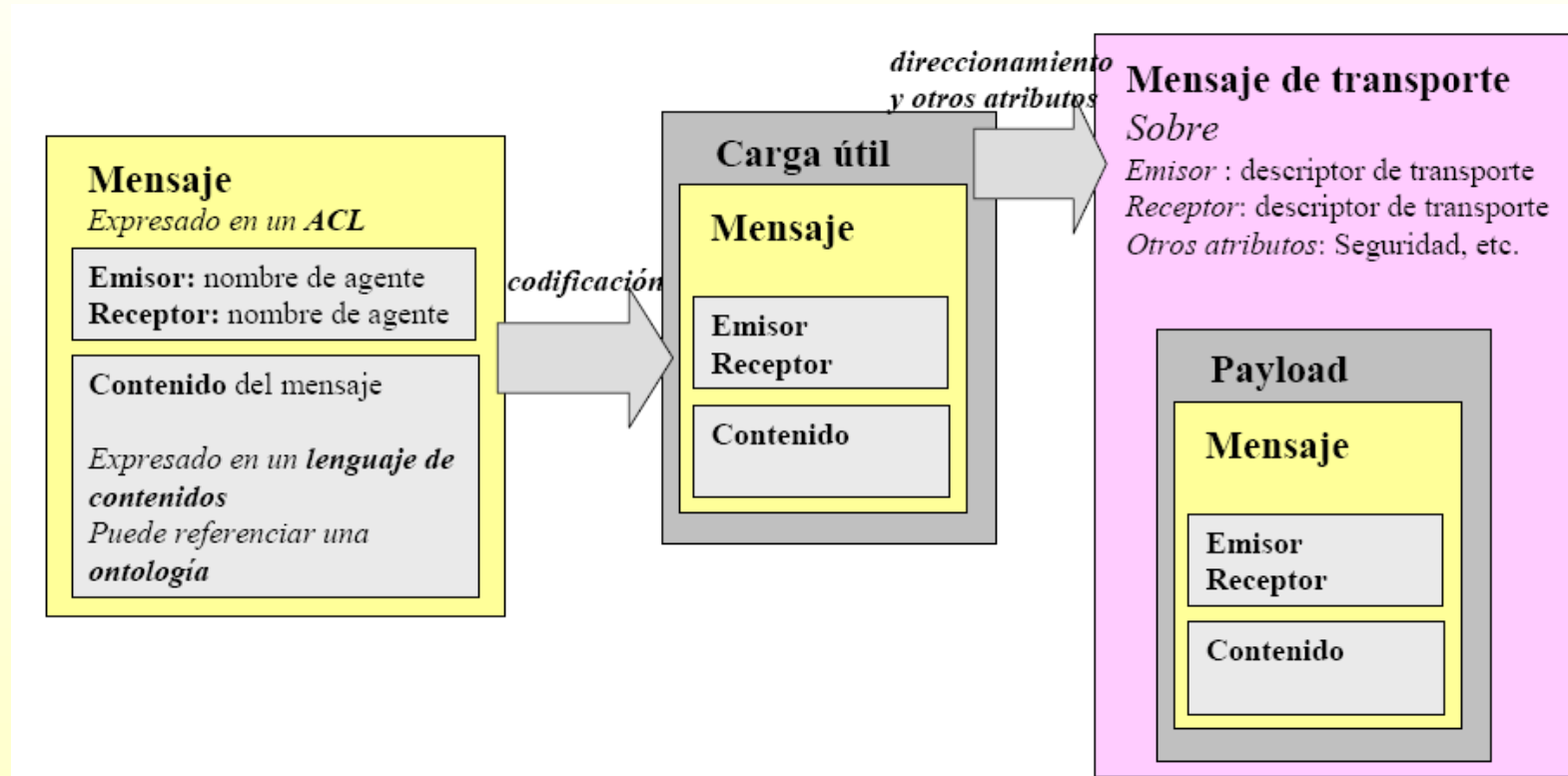
- Estructura de los mensajes
 - Los mensajes son tuplas clave-valor
 - Escritos en un lenguaje de comunicación de agentes (p.ej. FIPA ACL)
 - El contenido expresado con un lenguaje de contenidos (KIF, SL, ...)
 - El lenguaje de contenidos puede hacer referencia a una ontología
 - Incluyen los nombres de emisor y receptor
 - Un mensaje puede contener recursivamente otros mensajes
- Transporte de los mensajes
 - El mensaje de transporte consta de un campo de carga útil (payload) y un sobre (envelope)
 - El mensaje se transporta en la carga útil de un mensaje

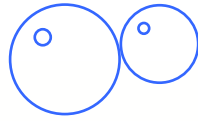




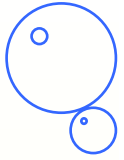
Arquitectura Abstracta FIPA

Phqvdlv#gh#r#djhq#w

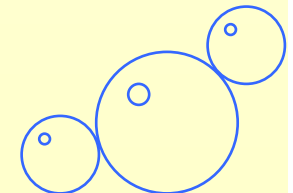
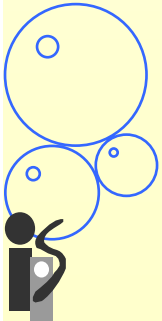


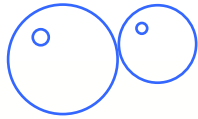


FIPA ACL

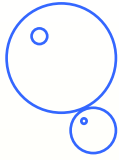


- Basado en actos del habla
 - **La semántica se basa en aptitudes mentales (creencias, intenciones, etc.)**
- Los mensajes son acciones comunicativas
 - **Sintaxis similar a KQML**
 - **Semántica formal definida con lógica modal**
- Además se definen protocolos de interacción de alto nivel, llamados conversaciones
- Es posible definir nuevas primitivas a partir de un núcleo de primitivas mediante composición
- El estado de los agentes se describe con el lenguaje SL (Semantic Language)



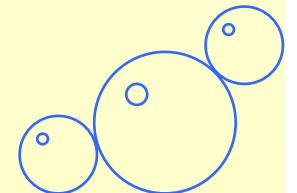
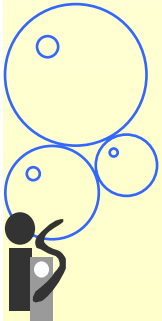


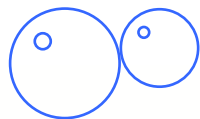
FIPA ACL



Ejemplo de mensaje ACL

```
#qirup
=vhqghudjhqwn4
=nhfhyhudjhqwn5
=frqwnq
      +suhfr#eur#1333,
=jqhhsq0wr uxqg07
=hsq0zlk eb37
=objxdjh vo
=qrwrj | qrwrdeur
/
```





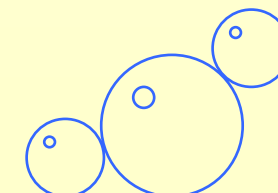
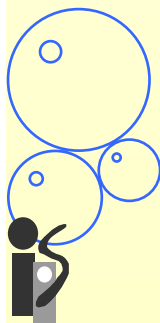
FIPA ACL

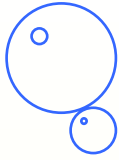
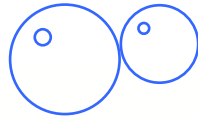
Parámetros del mensaje

=vhoqhu	=objxdjh
=hfhlyhu	=rqwrqj
=frqwhq	=hsqbe
=hsqbzlk	=surwfro
=bpúhsqbw	=froyhuvwlrq0g
=qyhrsh	

Tipos de mensaje

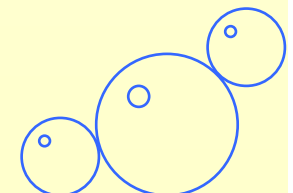
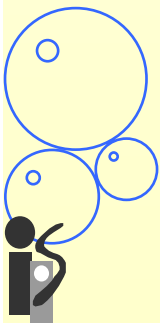
dfhsw@sursrvob	brup	uhxvh
djuh	brup 0li	uhfw@sursrvob
f0qfho	brup 0hi	untxhw
fis	qrw0qghuwrq	untxhw@zkhq
frqilp	sursrvh	untxhw@zkhqhyhu
gvrqilp	txhu 0li	vxevfueh
ibuh	txhu 0h	

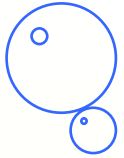
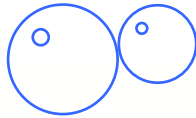




FIPA ACL

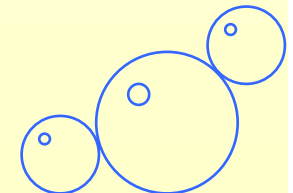
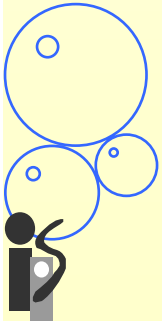
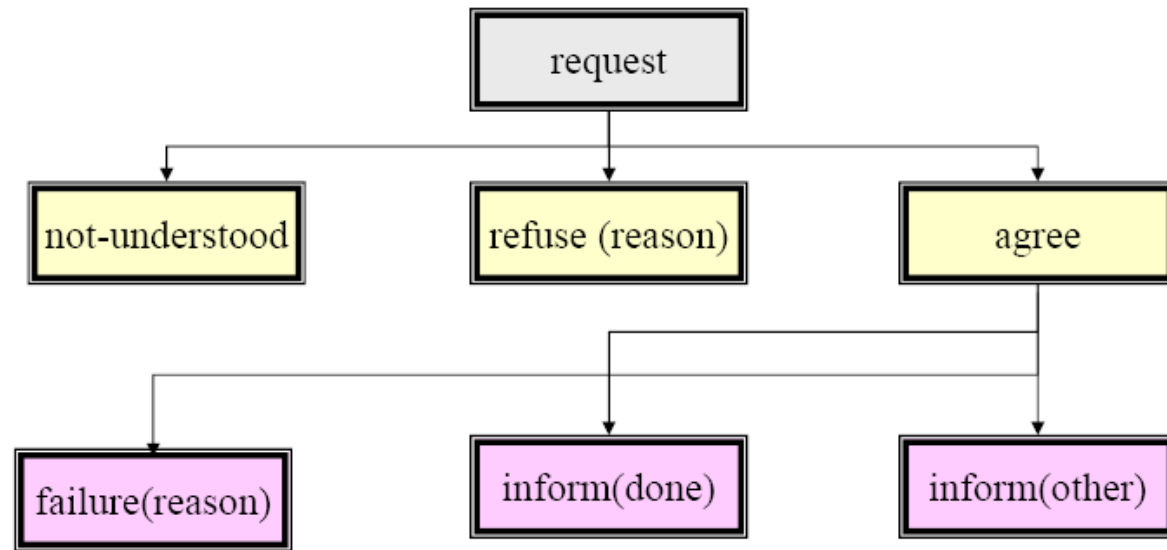
- Las conversaciones entre agentes suelen seguir unos ciertos patrones, secuencia típicas de mensajes: protocolos de conversación
- Un agente informa del protocolo que quiere usar mediante el parámetro :protocol
- Protocolos básicos definidos por FIPA:
 - **FIPA-request**
 - **FIPA-query**
 - **FIPA-request-when**
 - **FIPA-contract-net**
 - **FIPA-iteraterated-contract-net**
 - **FIPA-auction-english**
 - **FIPA-auction-ducth**



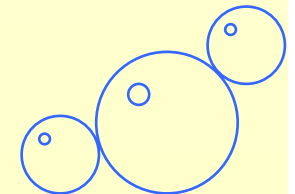
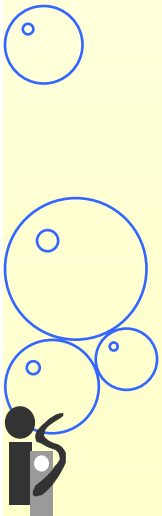
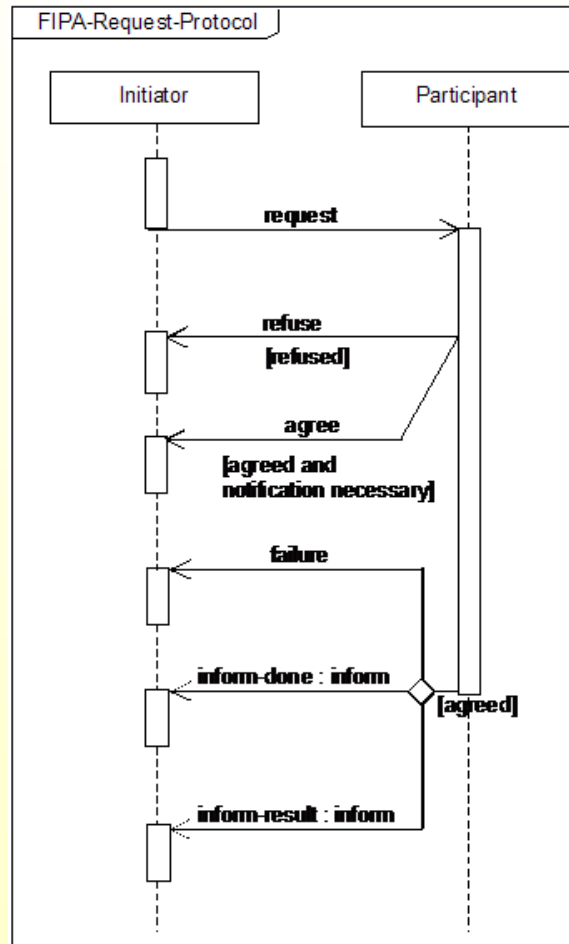


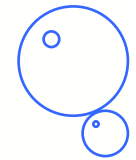
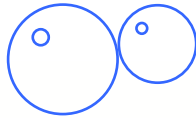
FIPA ACL

Protocolo FIPA request

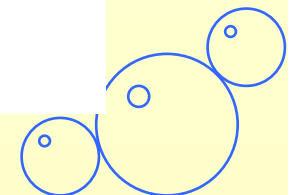
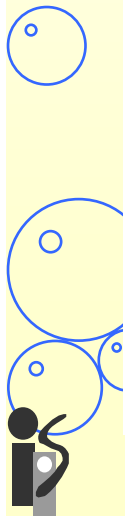
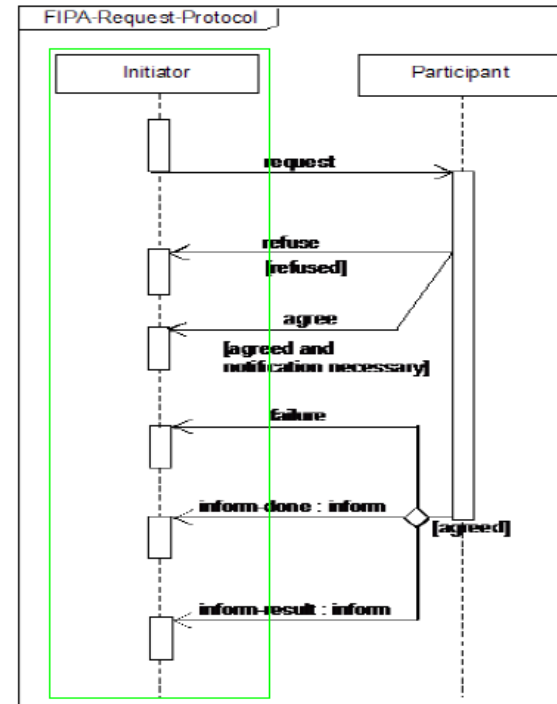
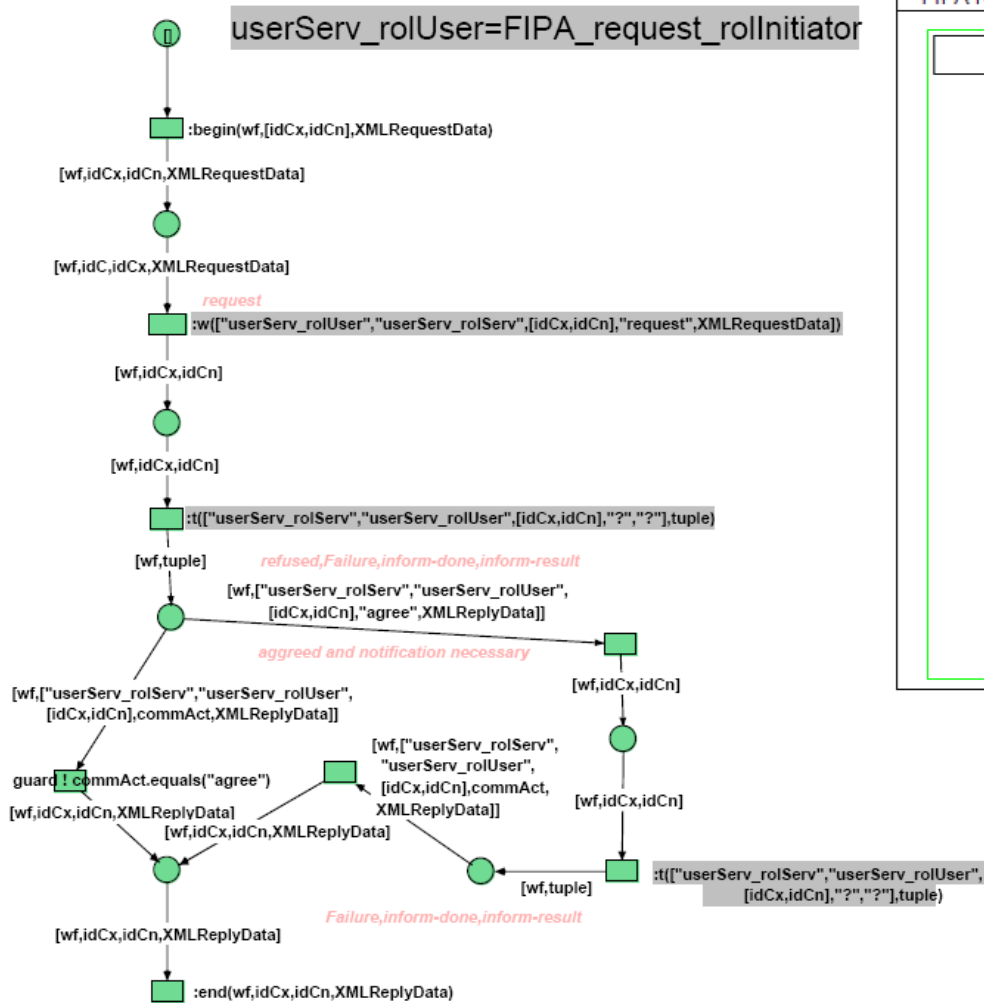


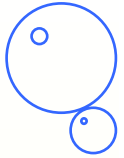
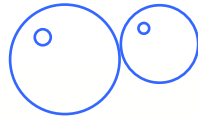
FIPA ACL





FIPA ACL (Renew/Linda Implementation)

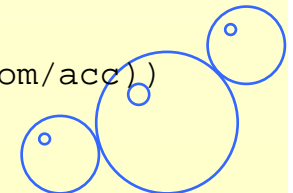
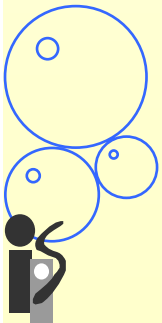




Ejemplos de diálogos

The agent dummy is created and it registers with the AMS of its home AP:

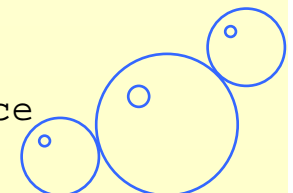
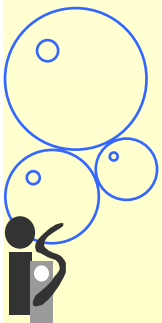
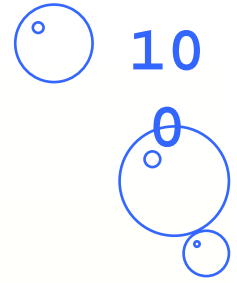
```
(request
  :sender
    (agent-identifier
      :name dummy@foo.com
      :addresses (sequence iiop://foo.com/acc))
  :receiver (set
    (agent-identifier
      :name ams@foo.com
      :addresses (sequence iiop://foo.com/acc)))
  :language fipa-sl0
  :protocol fipa-request
  :ontology fipa-agent-management
  :content
    "((action
      (agent-identifier
        :name ams@foo.com
        :addresses (sequence iiop://foo.com/acc))
      (register
        (ams-agent-description
          :name
            (agent-identifier
              :name dummy@foo.com
              :addresses (sequence iiop://foo.com/acc))
          :state active))))))")
```

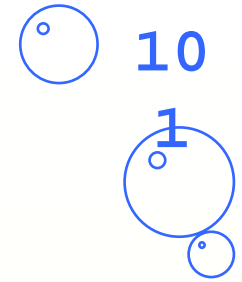
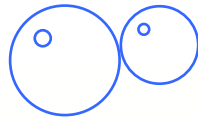


Ejemplos de diálogos

The AMS agrees and then informs dummy of the successful execution of the action:

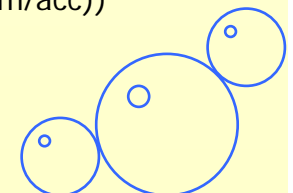
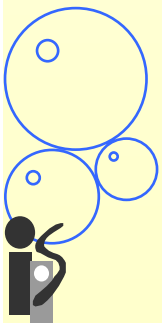
```
(agree
  :sender
    (agent-identifier
      :name ams@foo.com
      :addresses (sequence iiop://foo.com/acc))
  :receiver (set
    (agent-identifier
      :name dummy@foo.com
      :addresses (sequence iiop://foo.com/acc))
    :language fipa-s10
    :protocol fipa-request
    :ontology fipa-agent-management
  :content
    "((action
      (agent-identifier
        :name ams@foo.com
        :addresses (sequence iiop://foo.com/acc))
      (register
        (ams-agent-description
          :name
            (agent-identifier :name dummy@foo.com :addresses (sequence
              iiop://foo.com/acc)) :state active))) true)")
```

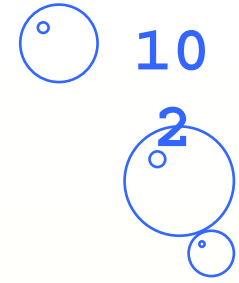
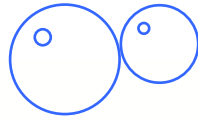




Ejemplos de diálogos

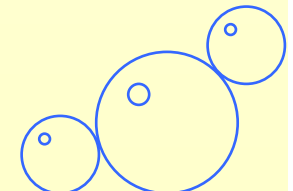
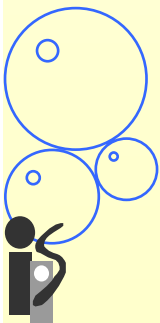
```
(inform
  :sender
    (agent-identifier
      :name ams@foo.com
      :addresses (sequence iiop://foo.com/acc))
  :receiver (set
    (agent-identifier
      :name dummy@foo.com
      :addresses (sequence iiop://foo.com/acc)))
  :language fipa-slo
  :protocol fipa-request
  :ontology fipa-agent-management
  :content
    "((done
      (action
        (agent-identifier
          :name ams@foo.com
          :addresses (sequence iiop://foo.com/acc))
        (register
          (ams-agent-description
            :name
              (agent-identifier
                :name dummy@foo.com
                :addresses (sequence iiop://foo.com/acc))
                :state active))))))")
```



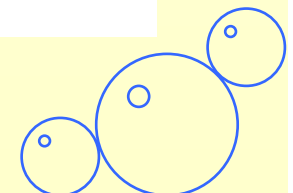
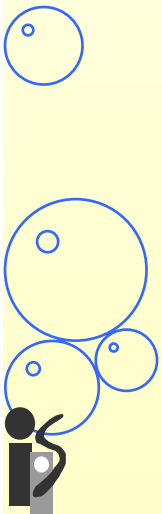
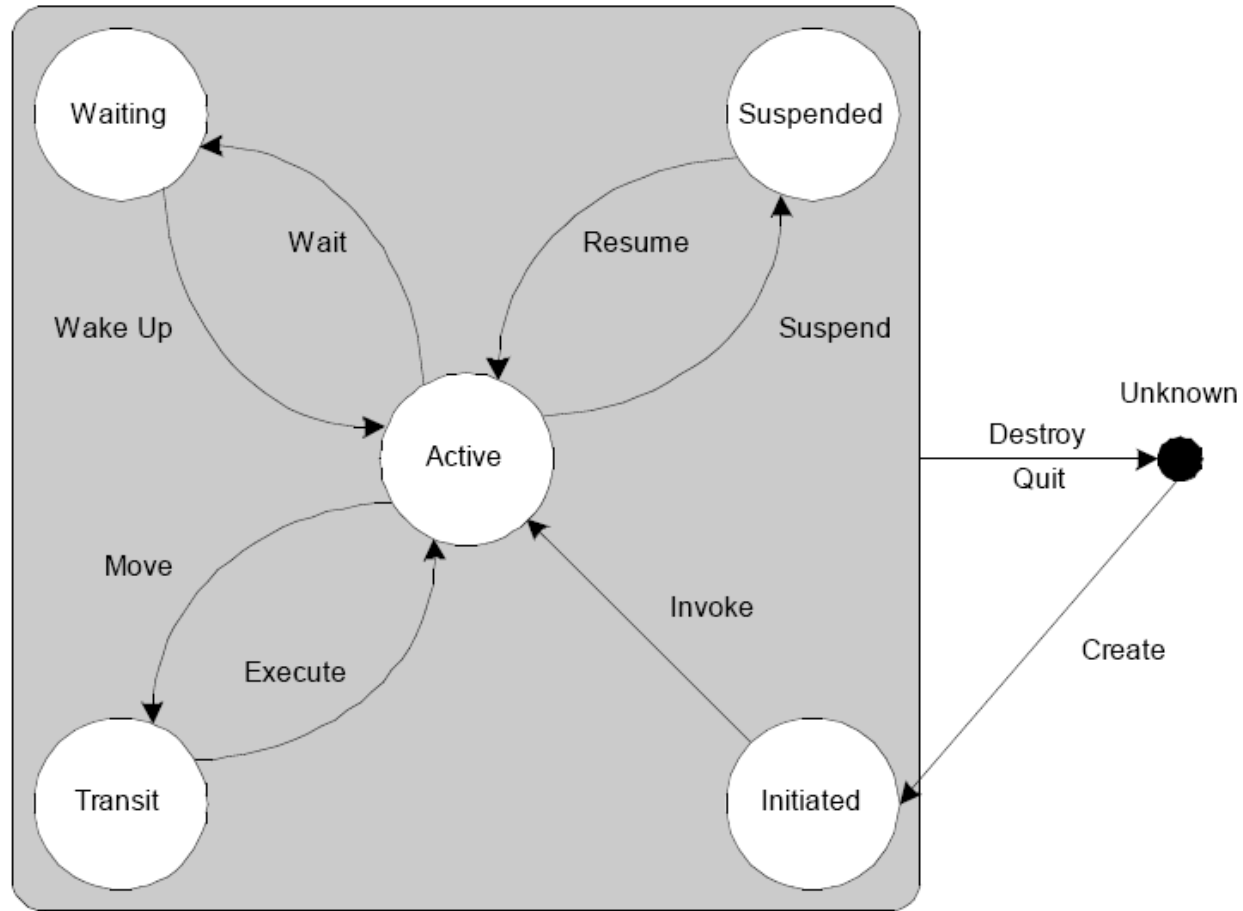


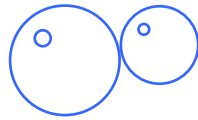
Referencias

- Brenner, W., Zarnekow, R. Wittig, H. *Intelligent Software Agents*. Springer, 1998.
- Briot J.P. y Demazeau, Y. *Principes et architecture des systèmes multi-agents*. Lavoisier, 2001.
- Ferber, J.: *Multi- Agent Systems*. Addison-Wesley. 1999.
- Gómez Sanz, J. Termostatos y Agentes, grasia.fdi.ucm.es/articulos/termostatos_agentes.rtf
- Jennings, N.R., *On agent-based software engineering*. Artificial Intelligence, 117, 2000.
- Russell, S., Norvig, P., *Artificial Intelligence: A Modern Approach*. Prentice-Hall, 1995.
- Shoham, Y. *Agent-oriented programming*. Artificial Intelligence, 60, 1993.
- Weiss, G. *Multiagent Systems*. MIT Press, 1999.
- Wooldridge, M.J., *Intelligent Agents* (capítulo del libro de Weiss)
- Ana MAS. Agentes Software y Sistemas Multiagentes. Conceptos, Arquitecturas y Aplicaciones. Prentice Hall, Juan Pavón, José L. Pérez de la Cruz Molina



Ciclo de vida de un agente gestionado por el AP





Referencias

- Bratman, M. E., Israel, D., and Pollack, M., Plans and Resource-bounded Practical Reasoning, *Journal of Computational Intelligence*, vol. 4, no. 4, pp. 349-355, 1988.
- Brenner, W., Zarnekow, R. Wittig, H. *Intelligent Software Agents*. Springer, 1998.
- Brooks, R.A., Intelligence without representation. *Artificial Intelligence*, 47, 1991.
- Carver, N. and Lesser, V. R.: *The Evolution of Blackboard Control Architectures*. Informe. Department of Computer Science, University Massachusetts. 1992
- Ferber, J. *Multiagent systems : an introduction to distributed artificial intelligence*, Addison-Wesley, 1999.
- Genesereth, M.R., Ketchpel, S.P. *Software Agents*. *CACM*, 37, 7, 1994
- Jennings, N.R., *On agent-based software engineering*. *Artificial Intelligence*, 117, 2000.
- Müller, J.P., *The design of intelligent agents: a layered approach*. *Lecture Notes in computer Science*, Vol.1177, Springer-Verlag, 1996
- Shoham, Y. *Agent-oriented programming*. *Artificial Intelligence*, 60, 1993.
- Sycara K.P. *Multiagent Systems*. *AI Magazine*, Summer 1998.
- Weiss, G. *Multiagent Systems*. The MIT Press, 1999.

