

# AI

## Inteligencia embebida y Objetos Inteligentes

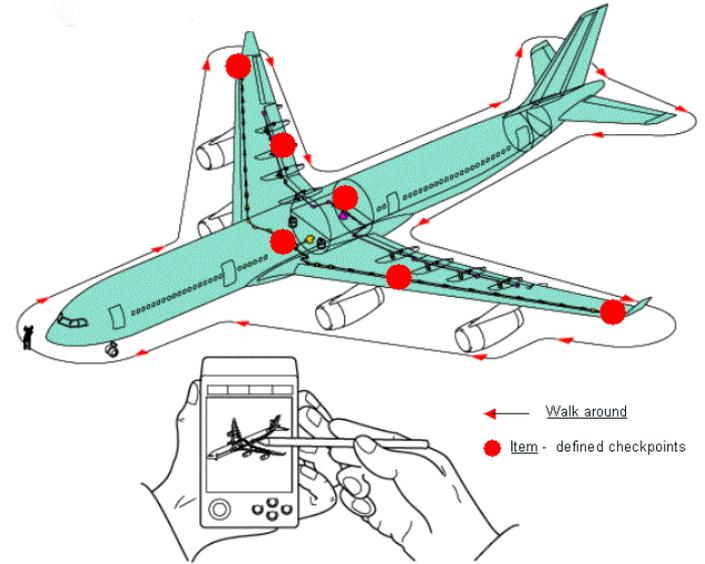
# Sensores



➤ Monitoreo individuo

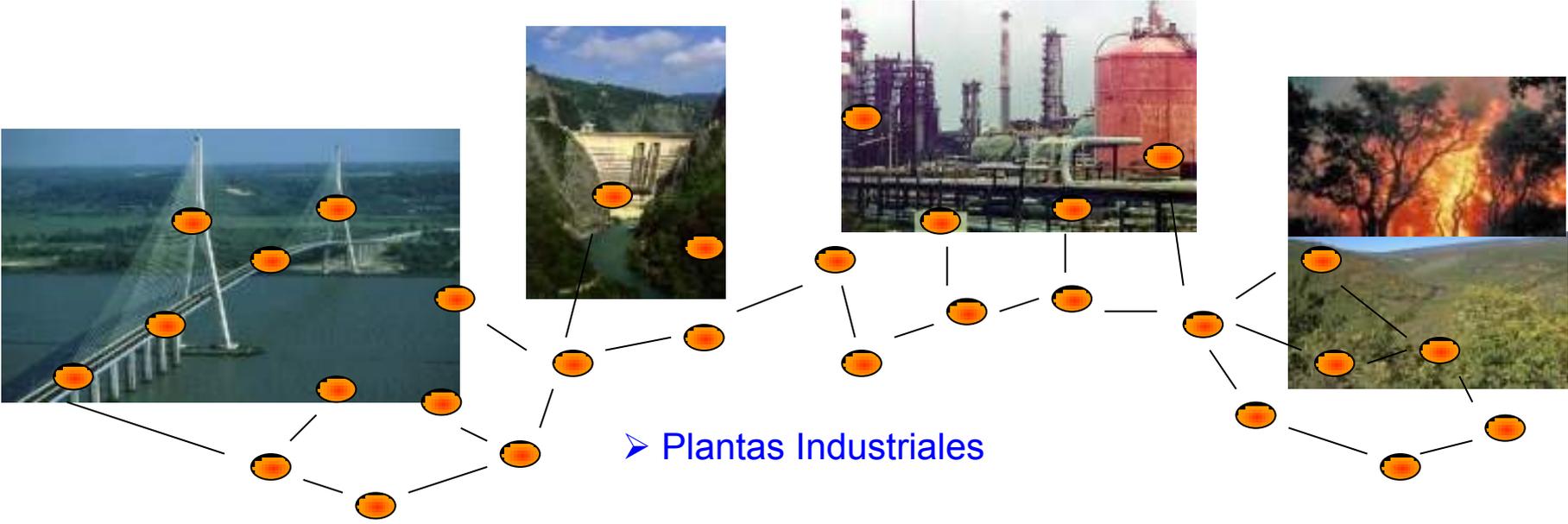


➤ sensores presión cauchos



➤ Monitoreo Avión

# Red de sensores

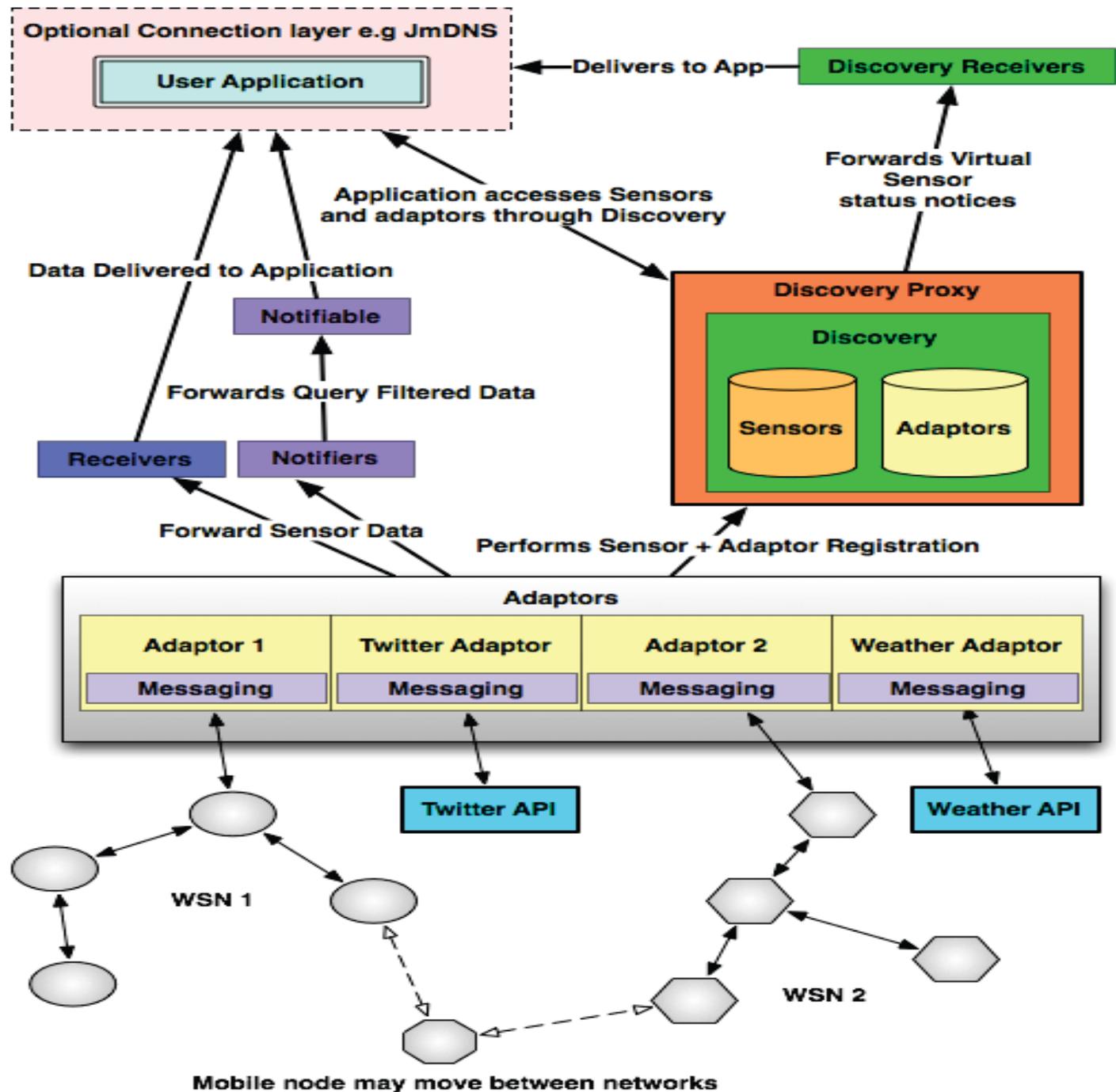


➤ Estructuras

➤ Plantas Industriales

➤ Fauna

# Middleware





# Objetos Inteligentes

## Imagínese el futuro:

- Usted transfiere una receta en un mensaje de texto con su celular a su ayudante de cocina “inteligente”, que, con la ayuda de etiquetas, determina la composición de los productos, sus fechas de vencimiento, etc.
- El ayudante le consulta a la nevera “inteligente”, la cual puede hacer otras cosas: generar alarmas por alimentos por vencerse, recordar medicamentos a tomar guardados en la nevera, etc.
- El ayudante envía al celular una lista de compras de los productos que faltan, y se necesitan para la receta. Incluso, el ayudante podría enviar una orden a su supermercado, de modo que las compras sean entregadas en su hogar, o estén listas para su recolección.

**La tecnología para alcanzar esto ya existe.**

**Sin embargo, la integración de los diversos dispositivos para combinar sus capacidades automáticamente, aun es una tarea pendiente.**

# Objetos Inteligentes

- Se carece de una apropiada metodología de diseño, capaz de abordar los problemas vinculados al hecho de ser artefactos autónomos, en un entorno cambiante.

**Se requiere de un paradigma donde los artefactos se auto-organicen, en el sentido que pueden comunicarse y permitir la emergencia de acciones imprevistas, para realizar tareas deseables, con la intervención mínima humana**

- El propósito no debe ser diseñarlos programarlos, o controlarlos, explícitamente. Los componentes deben obrar autónomamente entre sí y con el ambiente.

# Objetos Inteligentes

La auto-organización de los artefactos significa que diversos dispositivos, con diversas funcionalidades, en diversas configuraciones, pueden comunicarse e integrarse, para producir funcionalidades nuevas que los dispositivos solos no podrían alcanzar.

- Además, deben ser sistemas adaptativos y extensibles.
  - **Adaptativos** para hacer frente a los cambios inesperados en su ambiente, y
  - **flexibles** para tolerar acontecimientos para los que no fueron diseñados.



# Objetos Inteligentes

- Podemos dividir el **problema de integración en la emergencia**, en tres subproblemas:
  - Los dispositivos deben **aprender a comunicarse**, incluso cuando no tienen ninguna comprensión compartida a priori del problema;
  - Los dispositivos deben **aprender en quien pueden confiar** (otros dispositivos), para cooperar;
  - Los dispositivos deben desarrollar una **división eficiente del trabajo**, de modo que cada uno realice esa parte de la tarea en la que es más competente, mientras que delega las partes restantes a los otros.



# Objetos Inteligentes

- **Los dispositivos integrados deben no sólo comunicarse, sino cooperar.**

Por ejemplo, una persona en el aeropuerto quisiera que su teléfono móvil colabore con los dispositivos presentes en el aeropuerto, de modo que le digan, por ejemplo, cuando el vuelo está programado para salir, cuales son los salones VIP cercanos y su ruta, etc.

- **También deben aprender a coordinarse.**

Muchas de las tareas son mutuamente dependientes, por lo que deben aprender a coordinarse. La división del trabajo es un ejemplo, donde las tareas se delegan a los agentes “más expertos”.

**Cuando nuevos dispositivos se agregan al sistema se deben adaptarse mutuamente, produciendo una nueva organización.**

# Objetos Inteligentes



Veremos así en el futuro, una red de dispositivos especializados, que se auto-organizan, desde un proceso de aprendizaje entre ellos, de cómo eficazmente cooperar.

- Estos sistemas deberán ajustar sus capacidades, adaptándose a las tareas, a las situaciones, o a los dispositivos, encontrados.
- Esta capacidad de adaptarse debe estar ya presente en las reglas de interacción.
- La adaptación se puede alcanzar usando **retroalimentación**, tal que los agentes aprendan qué mensajes enviar, en quien pueden confiar, etc.

imaginarnos en nuestro futuro, dispositivos inteligentes, aprendiendo a ser más expertos, explorando.

# Objetos Inteligentes

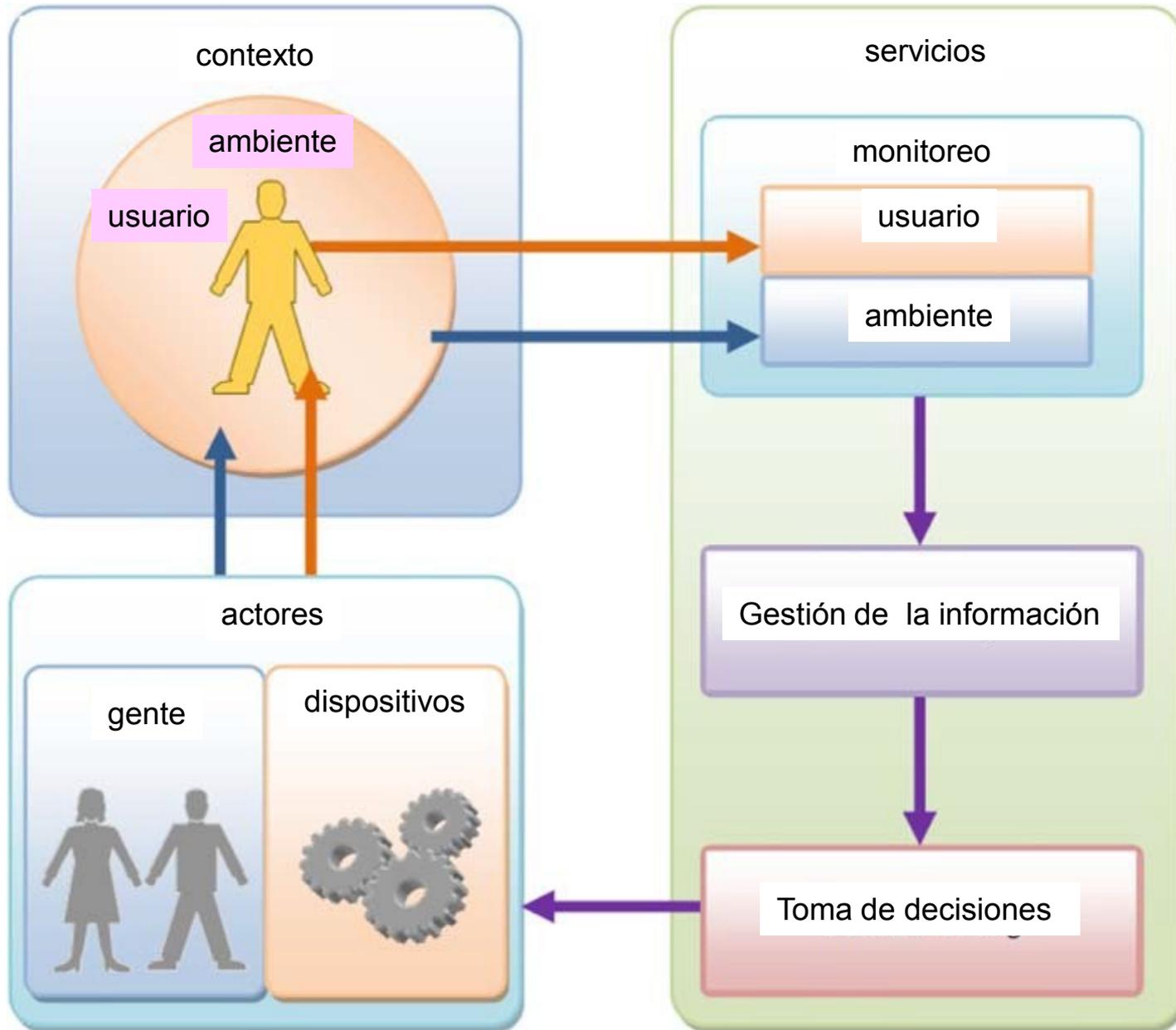


**No estamos hablando de la integración de muchas funciones en un solo dispositivo, sino de una red de dispositivos especializados, que se coordinan.**

- Esos dispositivos, obrando recíprocamente, aprenden sobre cómo comunicarse, con quienes cooperar, y cómo delegar y coordinar tareas especializadas.
- Pueden desarrollar una inteligencia colectiva, capaz de abordar tareas complejas.
- Serán sistemas en que la estructura y la función del sistema, “emergen” de las interacciones entre los dispositivos.

Los componentes obran libremente entre ellos y con el ambiente, adaptándose mutuamente, para alcanzar un estado “preferible” o configuración atractora.

# Entornos Inteligentes



# AI y teoría de Agentes

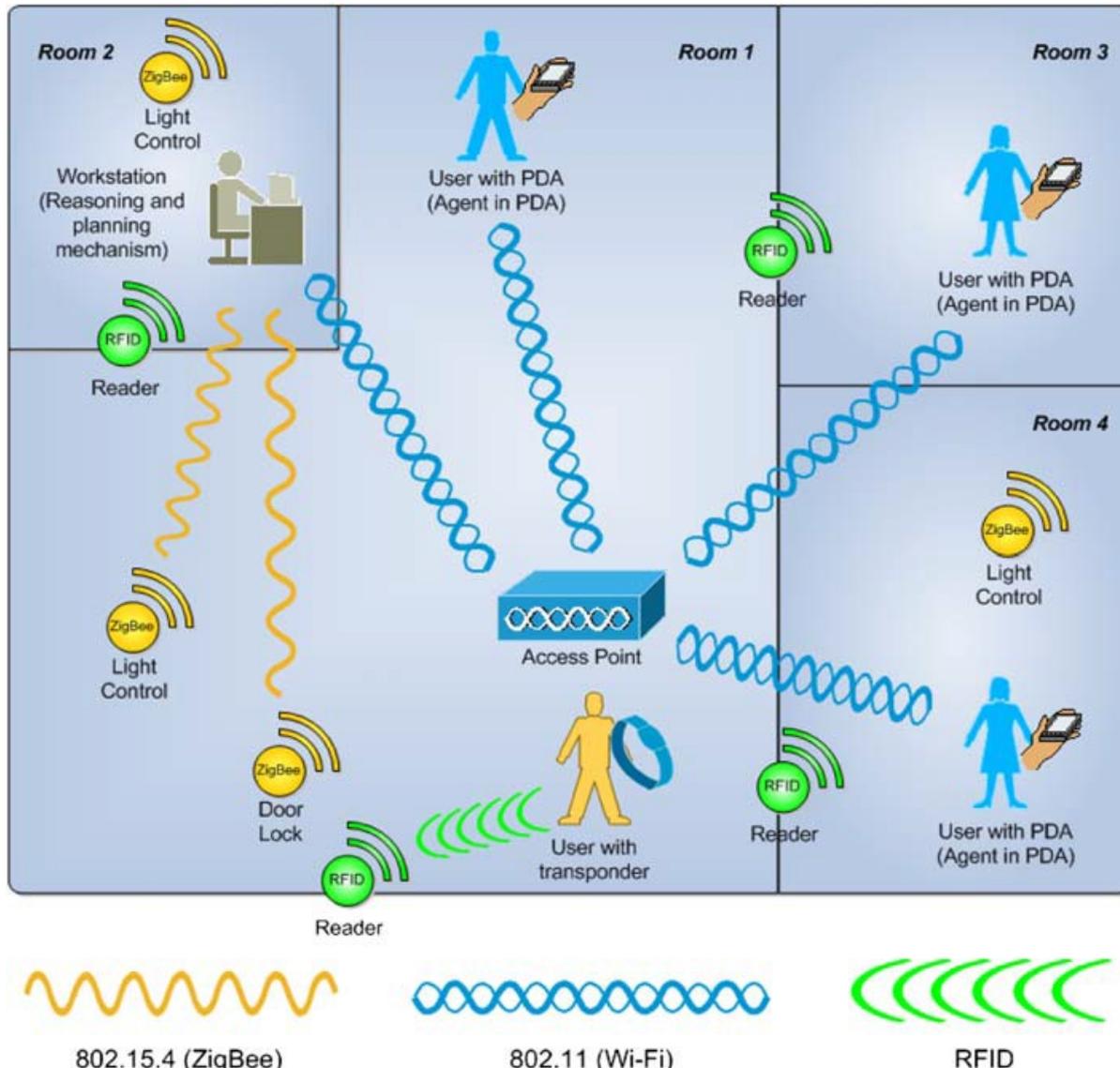
# Objetos Inteligentes

## Hacia los SMA



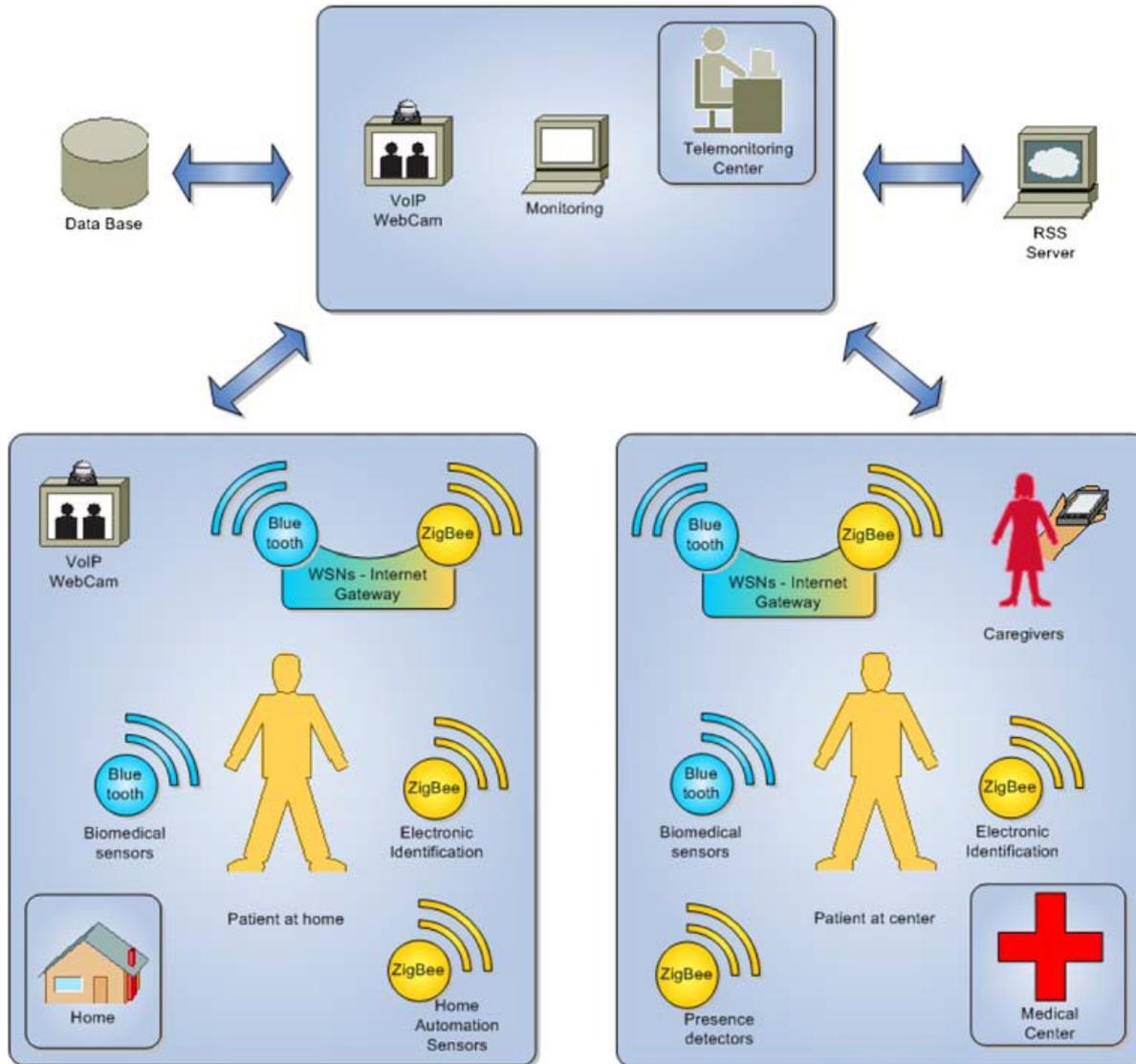
# Objetos Inteligentes

## Hacia los SMA



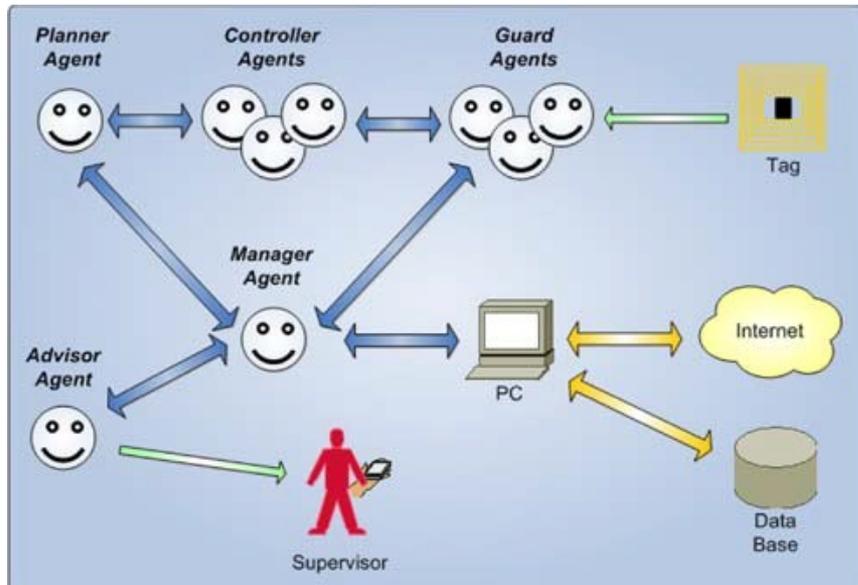
# Objetos Inteligentes

## Hacia los SMA



# Objetos Inteligentes

## Hacia los SMA





# Agente

Es cualquier cosa que pueda percibir su ambiente a través de *sensores*, y actuar sobre él a través de *actuadores*

- Agente Humano: ojos, oídos, manos, brazos, etc.,
- Agente Robot: cámaras, sensores y varios motores actuadores

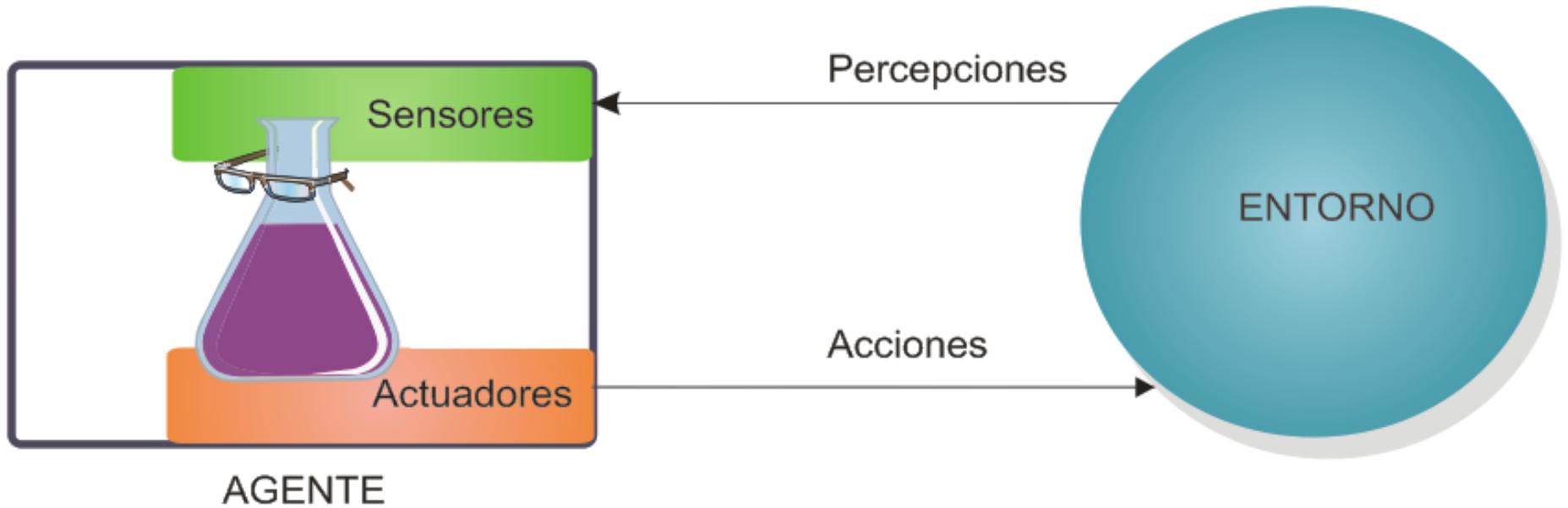


# Agente

Es un sistema computacional que está situado en un entorno, que es capaz de realizar acciones autónomas flexibles en ese entorno para alcanzar sus objetivos

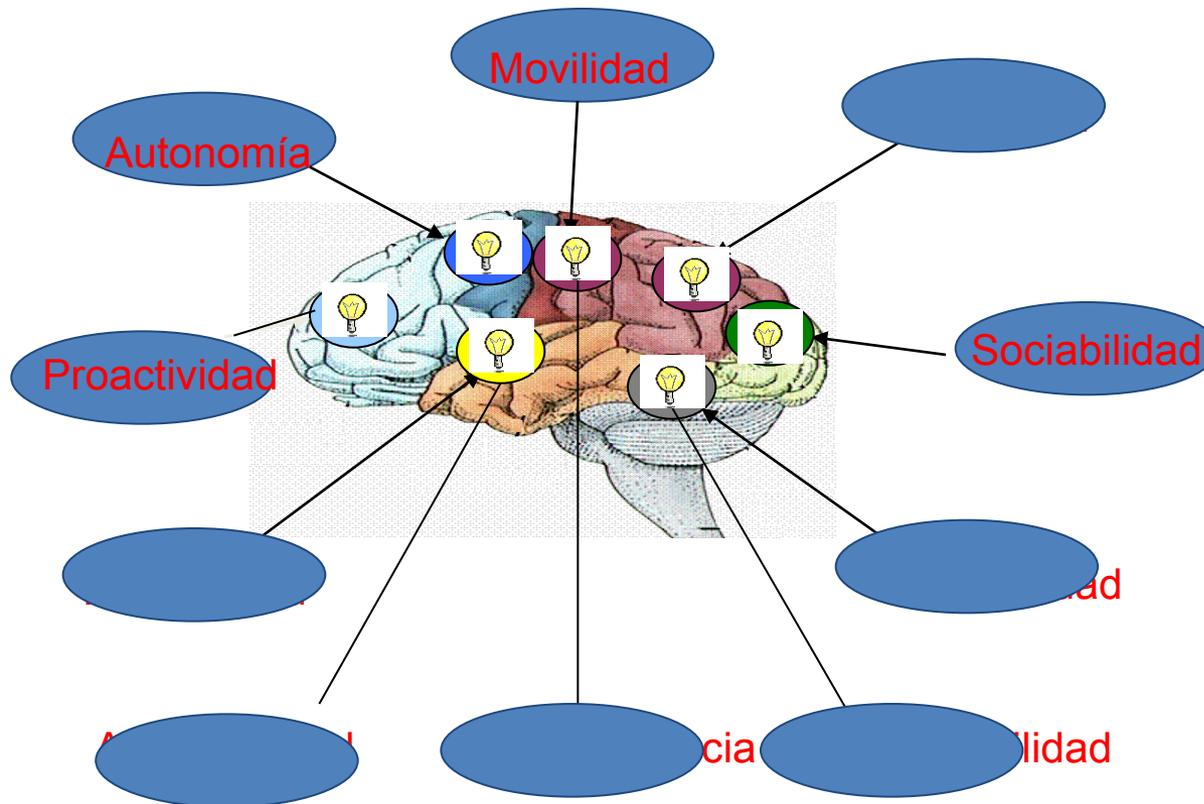
- Caracterizado por:
    - SU ESTRUCTURA (ARQUITECTURA)
    - SUS ACCIONES (COMPORTAMIENTO)
- Arquitectura+programa

# Agentes



# Agentes

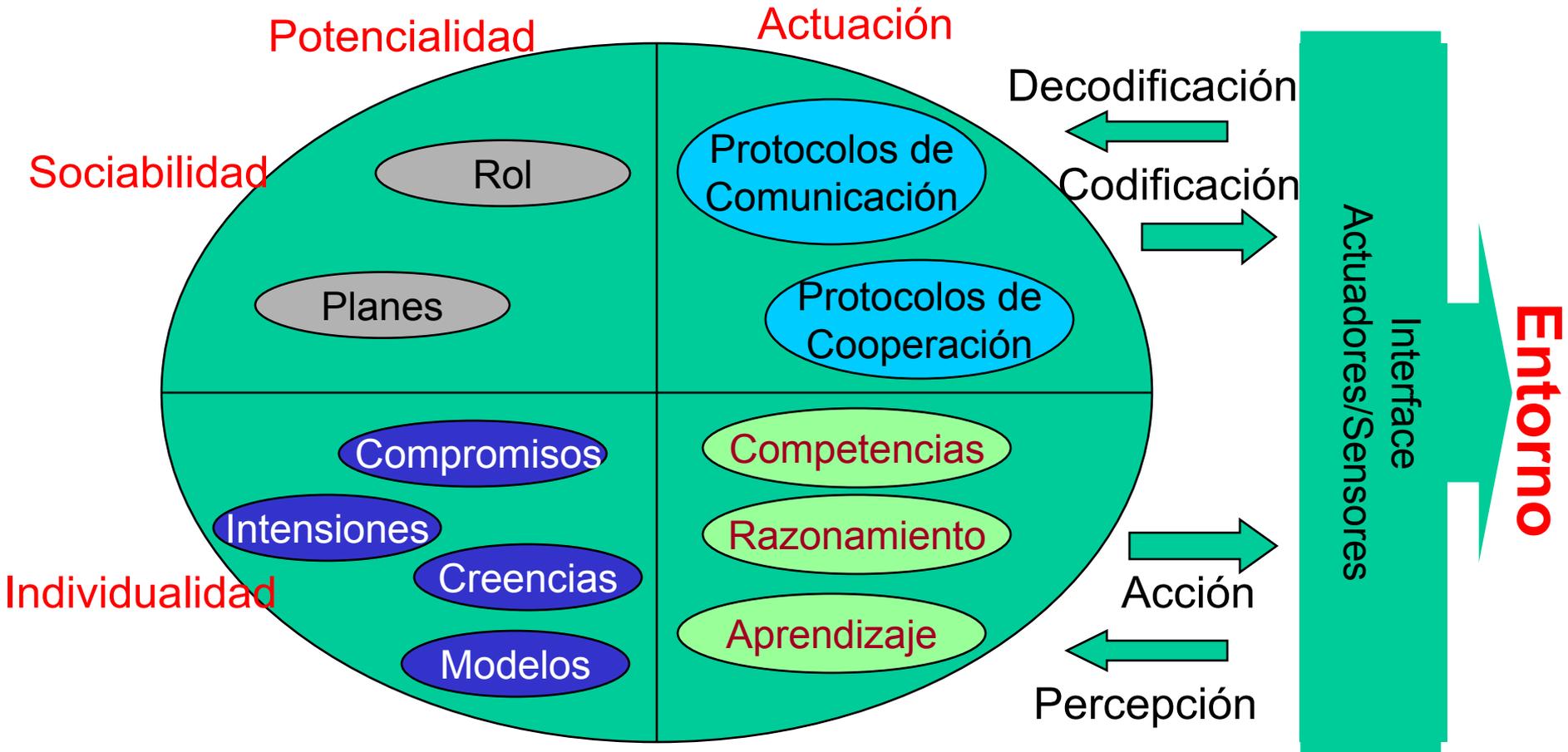
## Propiedades de un agente



# Otras propiedades de los Agentes

- Comunicación
- Veracidad
- Benevolencia
- Inteligencia
- Asincronismo

# Arquitectura de los Agentes



## Procedimiento de base:

1. Percibo (*Actualiza Memoria*)
2. Decido (*Escoge Acción* )
3. Actúo (*Actualiza Memoria*)

## Descripción práctica de un agente:

Sus Tareas.

Sus Conocimientos.

Su Comunicación



# Otros aspectos importantes a considerar en los Agentes

- Mecanismos para resolver un problema
- Mecanismo para planificar sus actividades /tareas
- Mecanismos para representar el conocimiento
- Mecanismo de razonamiento
- Mecanismos de aprendizaje
- Mecanismos de percepción
- Mecanismos para comunicarse

# SMA

- Son sistemas compuestos por múltiples componentes *autónomos* que poseen las siguientes características:
  - Cada agente tiene ciertas capacidades.
  - Necesitan de formas de coordinarse
  - Los datos no están centralizados
  - La computación es asíncrona.

SMAs	RPD
, Se enfocan primordialmente en la coordinación o negociación entre los agentes	Se enfocan en la resolución de un pb. dado, su descomposición, y síntesis de la soluciones
Son autónomos con la posibilidad de objetivos individuales	Necesitan sincronizarse
Permite la emergencia	Es determinista

# Características de ambientes multiagentes

- Proveen una infraestructura específica de comunicación y protocolos de interacción.
- Son típicamente abiertos y no tienen un diseño centralizado.
- Contienen agentes que son autónomos y distribuidos, y pueden ser interesados en si mismos o cooperar.

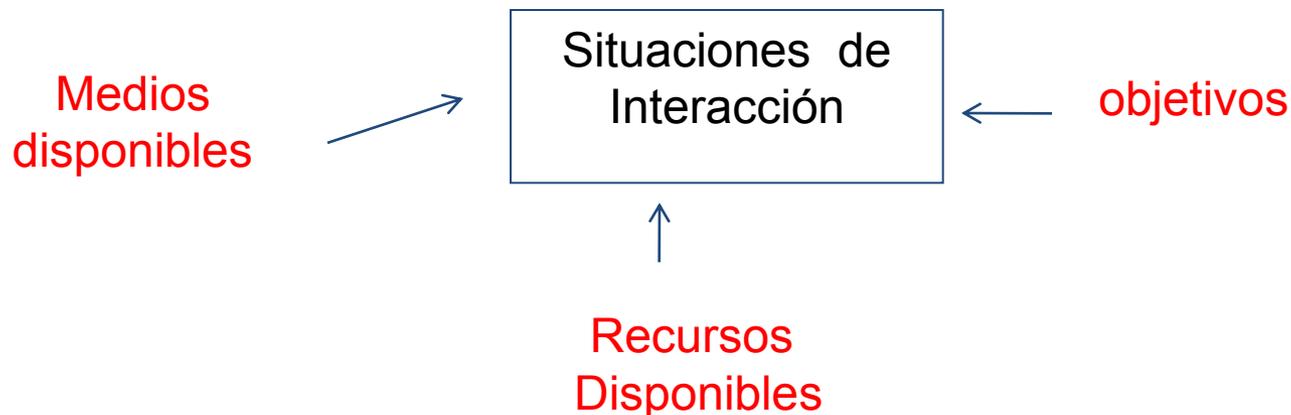
# Análisis Organizacional de los SMA

## IDENTIFICACION DE LAS PRINCIPALES FUNCIONES QUE LOS COMPONENTES DE UNA ORGANIZACIÓN DEBEN CUMPLIR

FUNCION\AMBITO	SOCIALES	RELACIONAL	AMBIENTAL	PERSONAL
<b>REPRESENTACION</b>	REPRESENTACION DE LA SOCIEDAD (ORGANIZAC.) DONDE ESTA INMERSO: ROLES, FUNCIONES,...	REPRESENTACION DE LOS OTROS (ENTRE AGENTES, O CON OTRAS ORGANIZAC.): OBJETIVOS, COMPETENCIA., ETC.	REPRESENTACION DEL MUNDO, LEY DEL UNIVERSO	AUTO-REPRESENTACION
<b>ORGANIZARSE</b>	PLANIFICACION. ACCIONES SOCIALES	COORDINACION COMUNICACIONES, PLANIF.ICACION INTERACCIONES	PLANIFICACION ACCIONES Y CONTROL DE EJECUCION EN EL AMBIENTE	META-PLANIFICACION, GESTION TAREAS INTERNAS
<b>ELEMENTOS PARA TOMA DE DECISION</b>	OBJETIVOS Y RESTRICCIONES COLECTIVAS DE LA SOCIEDAD (ORGAN.)	DEMANDAS Y RESTRICCIONES DE LOS OTROS	FUENTES DE DESEOS, PLACER, ETC. QUE EXISTEN EN EL AMBIENTE	NECESIDADES Y RESTRICCIONES INTERNAS.
<b>INTERACCION</b>	PRIMITIVAS DE INTERACCION EN LA SOCIEDAD (ORGAN.)	MECANISMOS DE COMUNUNICACION CON LOS OTROS	MECANISMOS DE PERCEPCION Y ACCION	AUTO-COMUNICARSE, AUTO-HABLARSE
<b>PRODUCCION U OPERATIVAS</b>	GESTION. Y DIRECCION EN LA SOCIEDAD (ORGAN.)	COORDINACION Y COOPERACION CON LOS OTROS.	CONTROL, DIAGNOSTICO, ETC SOBRE AMBIENTE	AUTO-MODIFICACION (P.E. APRENDIZ.AJE), AUTO-GENERARSE, ...
<b>CONSERVACION</b>	CONSERVACION DE LA SOCIEDAD (ORGANIZ.)	CONSERVACION DE LAS RELACIONES Y DE LOS OTROS.	CONSERVACION RECURSOS DEL AMB., DEFENSA	CONSERVACION INDIVIDUAL, REPARACION

# Interacciones

Una situación de **interacción se produce cuando dos o más agentes, por medio de una serie de acciones recíprocas** confluyen en una **relación dinámica**. Estas acciones, las influencias que unos ejercen sobre los otros, las obligaciones y compromisos que establecen, hacen de los agentes **entidades sociales**.



# Interacciones

---

- La interacción en un SMA está determinada por una serie de protocolos de comunicación que establecen los *actos de habla* válidos que permiten intercambiar información entre los agentes para lograr un objetivo.
- Entre los protocolos más utilizados se pueden nombrar el de Solicitud (Request), Consulta (Query), Redes de Contrato (Contract-Net), Subasta Inglesa (British Auction) y Subasta Holandesa (Dutch Auction).

# INTERACCIONES

OBJETIVOS	RECURSOS	CAPACIDADES	TIPO	CATEGORIA
COMPATIBLES	SUFICIENTES	SUFICIENTES	<i>INDEPEND.</i>	INDIFERENTE (PERS. CAMINA ACERA)
COMPATIBLES	SUFICIENTES	INSUFICIENTES	<i>COLABOR. SIMPLE</i>	COOPERACION (ASIGNACION DE TAREAS)
COMPATIBLES	INSUFICIENTES	SUFICIENTES	<i>ENTRABAR</i>	COOPERACION (CIRCULAC. CARROS, USO PROCESADOR, TRAF. AER)
COMPATIBLES	INSUFICIENTES	INSUFICIENTES	<i>COLABOR. COORD.</i>	COOPERACION (FABRICACION INDUSTRIAL)
INCOMPATIBLES	SUFICIENTES	SUFICIENTES	<i>COMPET. INDIV. PURA</i>	ANTAGONISMO (MARATON)
INCOMPATIBLES	SUFICIENTES	INSUFICIENTES	<i>COMPET. COLECT. PURA</i>	ANTAGONISMO (CARRERA DE RELEVO)
INCOMPATIBLES	INSUFICIENTES	SUFICIENTES	<i>CONFLICTO INDIV. RECUR.</i>	ANTAGONISMO (DEFENSA TERRITORIO, JEFE MANADA)
INCOMPATIBLES	INSUFICIENTES	INSUFICIENTES	<i>CONFLICT. COLECT RECUR.</i>	ANTAGONISMO (GUERRAS, COMPET. INDUSTRIAL)

# Tipos más comunes de interacción

- **Coordinación:** propiedad de interacción entre un conjunto de agentes que realizan alguna actividad colectiva:
  - **Coordinación global:** el sistema determina y planifica globalmente, existiendo un agente que verifica todos los conflictos.
  - **Coordinación individual:** cada agente tiene completa autonomía para decidir qué hacer y resolver los conflictos que detecte localmente.
- **Cooperación:** una clase de coordinación que consiste en que varios agentes no antagonistas (acciones de uno no perjudiquen al otro) interactúen entre sí con el objetivo de *conseguir un fin común*.
- **Negociación:** consiste en poner de acuerdo a los agentes de un sistema, cuando cada uno defiende sus propios intereses, llevándolos a una *situación que los beneficie a todos* (comercio electrónico, subastas, ...)

# Interacción indirecta

- Las acciones que toman los agentes *modifican su ambiente o su estado interno*.
- Los otros agentes llegan a *observar estos cambios que afecta su comportamiento posterior*.
- Modelo matemático:
  - *x es un agente y  $x_t$  es su estado en tiempo t*
  - *F(x) es el conjunto de agentes que x está observando*
  - *f() es una función que gobierna los cambios del estado interno de x*
  - *g() es una función que modela el efecto de las observaciones de x sobre los demás!*
  - *$x_t = 1$  significa que el agente x emite un señal al tiempo t*

$$x_{t+1} = \begin{cases} f(x_t) + g(F(x), t) & \text{Si } \exists y \in F(x) \text{ tal que } y_t = 1 \\ f(x_t) & \text{de lo contrario} \end{cases}$$

# Tipos de Interacción

**Cooperación**

**Coordinación**

**Negociación**

# COOPERACION

***proceso por el que ciertos agentes participantes generan deberes mutuamente dependientes para actividades conjuntas (planes).***

Las situaciones de cooperación aparecen cuando los agentes tienen que resolver problemas o tareas interdependientes

- Se recibe un problema en cierto nivel de abstracción.
- El agente resuelve localmente aquello que es posible.
- Recurre a otros agentes del mismo nivel para el resto de las tareas.
- Recurre a otros niveles de abstracción para el resto de las tareas.

# COOPERACION

## A. FORMAS DE COOPERACION

- COOPERACION COMO ACTITUD INTENCIONAL
- COOPERACION DESDE EL PUNTO DE VISTA DEL OBSERVADOR

## B. INDICES PARA CALIFICAR LA COOPERACION

COORDINACION DE ACCIONES

COMPARTIR RECURSOS

GRADO DE PARALELISMO

ROBUSTES

REDUNDANCIA DE LAS ACCIONES

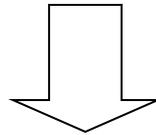
PERSISTENCIAS DE LOS CONFLICTOS

# COLABORACION Y REPARTICION DE TAREAS

Repartir tareas, recursos e información implica:

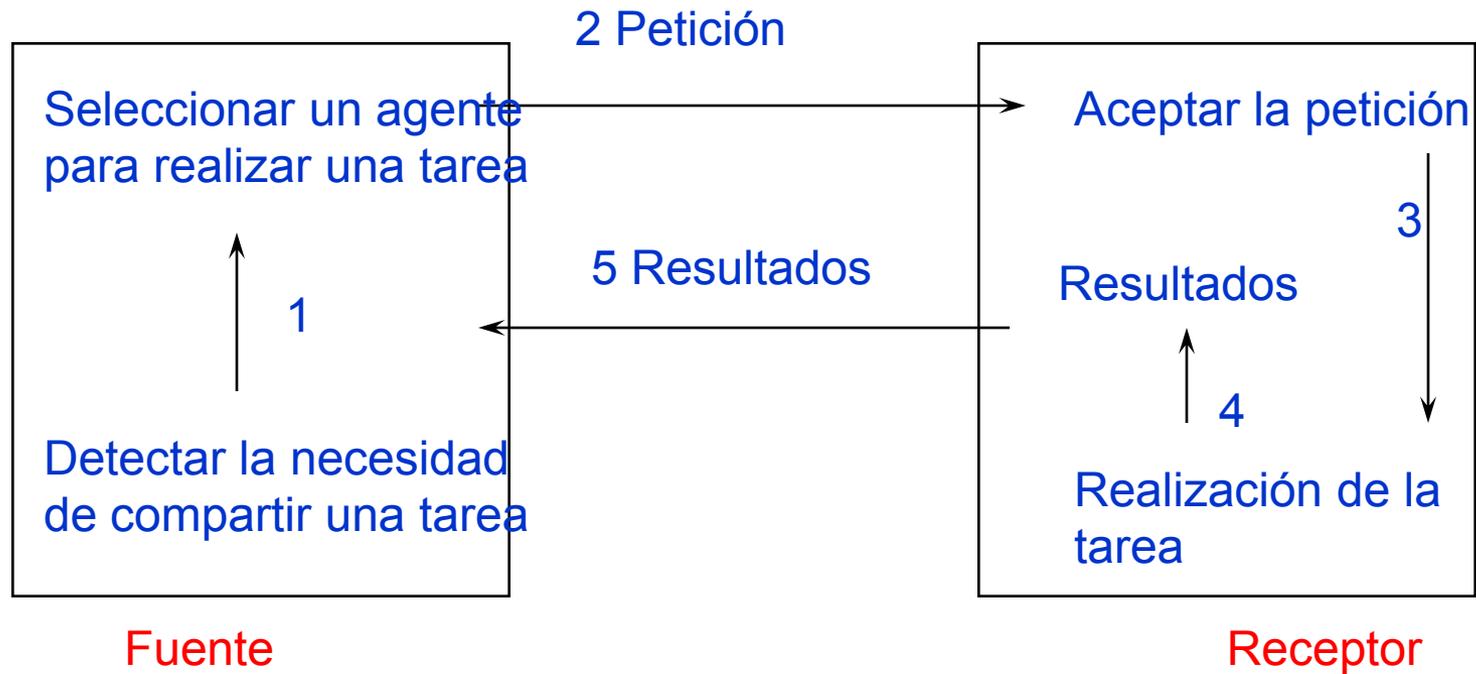
- Quién debe hacer qué con qué medios

En función de los objetivos y competencias de los agentes y restricciones del ambiente



Descomposición de Tareas

# REPARTICION DE TAREAS



# COLABORACION Y REPARTICION DE TAREAS

- ROLES

CLIENTES O DEMANDADORES

PROVEEDORES O SERVIDORES

MEDIADORES

- FORMAS DE ASIGNACION

PREDEFINIDA

CENTRALIZADA

- JERARQUICO=> IMPUESTA

- IGUALITARIO => MEDIADOR

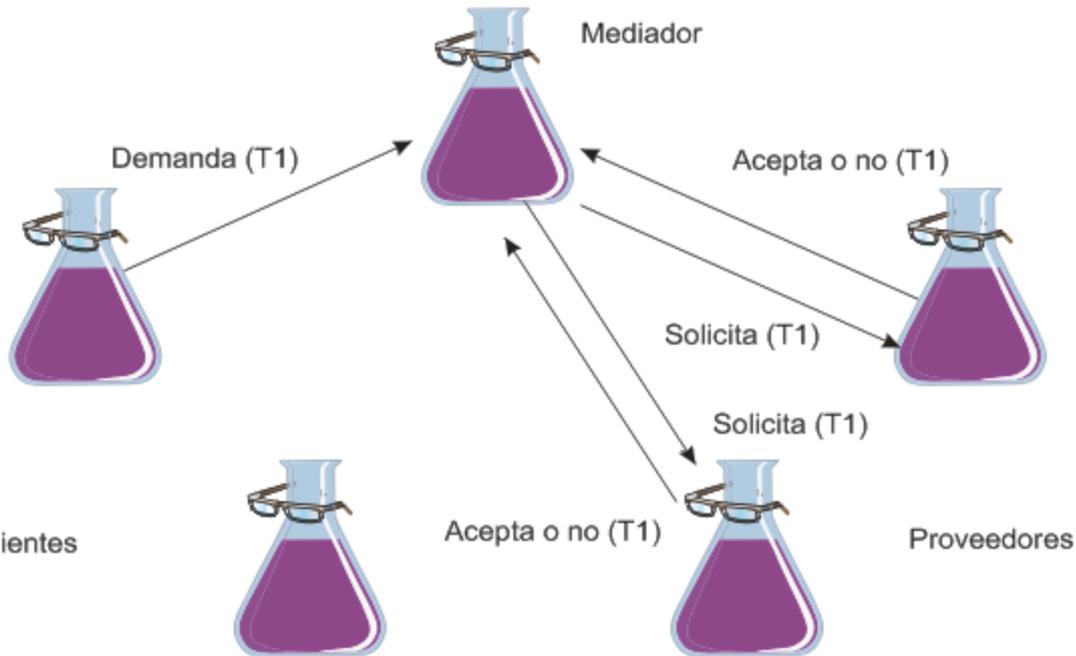
DISTRIBUIDA

- RED DE CONTACTOS: directa o delegar

- RED DE CONTRATO: licitación o subasta

EMERGENTE

# Asignación de Recursos



Centralizada clásica.  
Centralizada con mediador.  
Por redes de amistades.  
por concurso  
Emergente

# ASIGNACION ESTATICA CON MEDIADOR

- *ESTRUCTURA DE DATOS*
  - LISTA DE AGENTES CON SUS COMPETENCIAS
  - DEMANDA DEL MEDIADOR (TAREA A HACER SOLICITADA)
  
- *DOS CASOS*
  - MEDIADOR CONOCE FUNCION DE EVALUACION DEL PROVEEDOR
  - MEDIADOR NO CONOCE FUNCION DE EVALUACION (CADA PROVEEDOR DEBE FORMULAR UNA PROPUESTA)

# COORDINACION DE ACCIONES

## ***CONJUNTO DE ACTIVIDADES SUPLEMENTARIAS QUE SE DEBEN REALIZAR EN UN SMA***

Tareas no directamente productivas que mejoran la eficiencia total del sistema. Los agentes se coordinan para gestionar las dependencias existentes entre sus actividades

- ARTICULACION DE ACTIVIDADES HECHAS POR CADA AGENTE
- MEJORAR ACCION DEL GRUPO:
  - **AUMENTO DE RENDIMIENTOS**
  - **DISMINUCION DE CONFLICTOS**

# COORDINACION DE ACCIONES

- POR QUE COORDINAR:
  - AGENTES NECESITAN RESULTADOS E INFORMACION PRODUCIDAS POR OTROS AGENTES
  - RECURSOS SON LIMITADOS
  - OPTIMIZAR COSTOS
  - PONER A TRABAJAR JUNTOS A AGENTES CON OBJETIVOS DIFERENTES PERO QUE DEPENDEN ENTRE SI

# COORDINACION DE ACCIONES

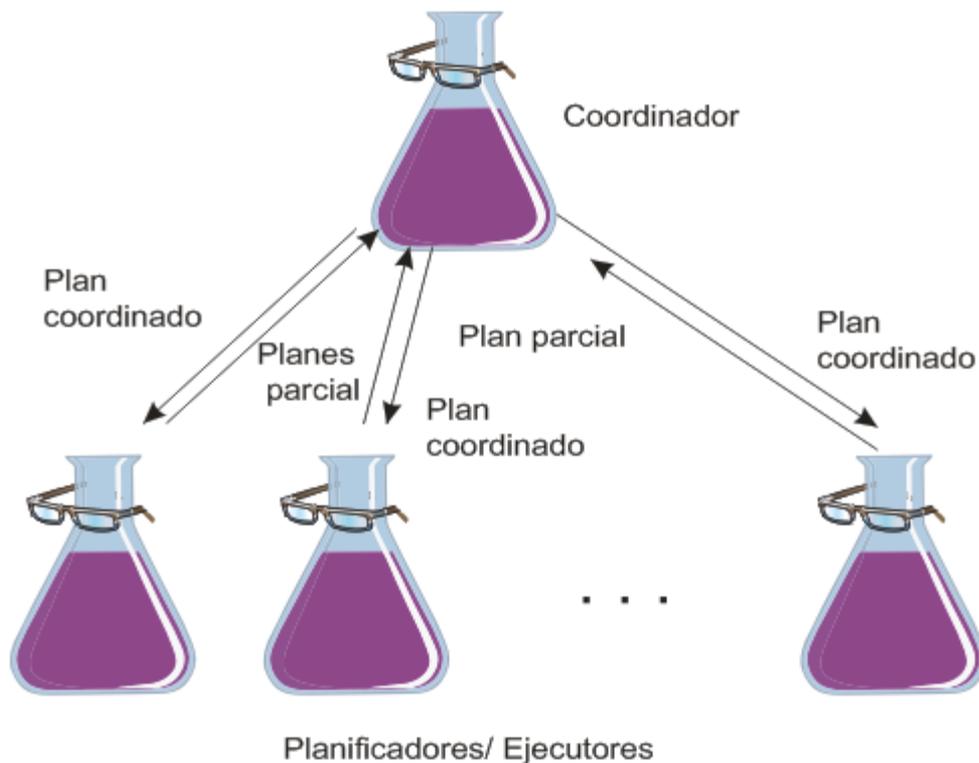
Dos enfoques para su solución:

- **Centralización**
  - Un agente coordinador especial es el responsable de detectar las interdependencias entre las actividades de los agentes locales.
- **Descentralización**
  - Los agentes interactúan entre sí. Poseen el conocimiento para descubrir inconsistencias entre sus acciones previstas y adaptar mutuamente sus decisiones locales.

# FORMAS DE COORDINACION DE ACCIONES

- SINCRONIZACION:
  - DE MOVIMIENTOS
  - DE ACCESO A RECURSOS => SECCIONES CRITICAS
- POR PLANIFICACION
  - PLANIFICACION CENTRALIZADA VS. DISTRIBUIDA
  - PLAN CENTRAL VS. DISTRIBUIDO
- REACTIVA
  - ACCIONES SITUADAS (CAMPOS DE POTENCIAS , RECURSO=>SINCRONIZACION)
  - MARCANDO EL AMBIENTE
  - DE MANADA/O JAURIA (AGREGAR, DISPERSAR, EVITAR, VIGILAR)
- POR REGLAMENTACION
  - REGLAS SOCIALES

# Planificación



Planificación centralizado para planes distribuidos

Coordinación centralizada para planes parciales

Planificación distribuida de un plan centralizado

Planificación distribuida para planes distribuidos

Jerarquización de planes.

Planificación reactiva

# PLANIFICACION MULTIAGENTES

- PLANIFICACION CENTRALIZADA PARA AGENTES MULTIPLES:
  1. REALIZACION DE UN PLAN GENERAL (EL PLANIFICADOR)
  2. IDENTIFICACION RAMIFICACIONES QUE SE PUEDEN REALIZAR EN PARALELO (POP)
  3. ASIGNACION (ESTATICA O DINAMICA) DE LAS TAREAS A LOS AGENTES
- COORDINACION CENTRALIZADA POR PLANES PARCIALES
  1. CADA AGENTE HACE SU PLAN PARCIAL
  2. EL COORDINADOR FUSIONA LOS PLANES (RELACION ENTRE LAS ACCIONES)
- PLANIFICACION DISTRIBUIDA
  1. CADA AGENTE HACE SU PLAN PARCIAL
  2. NEGOCIACION

# Negociación

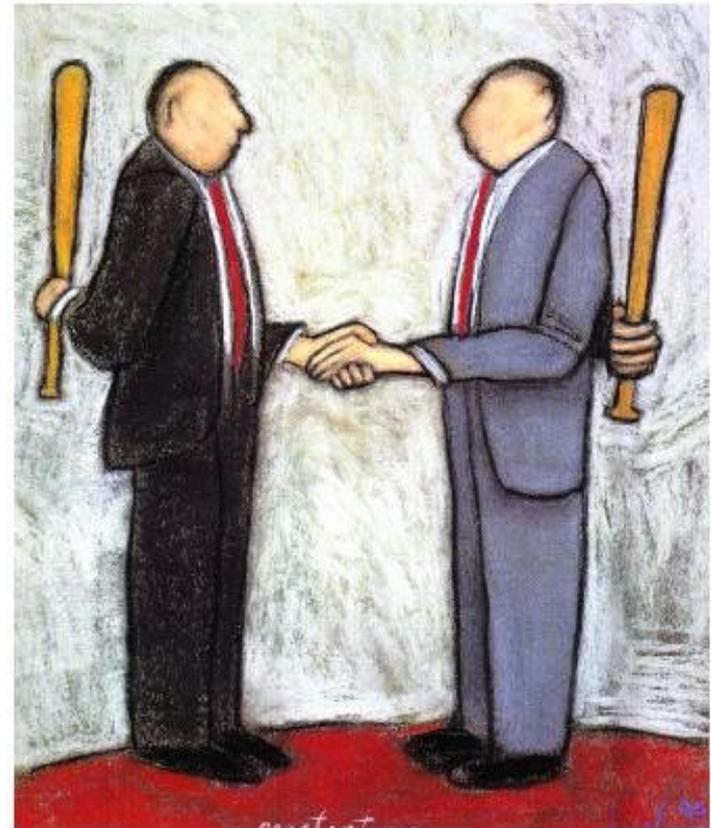
**Forma de interacción en la cual un grupo de agentes, con intereses conflictivos y un deseo de cooperar, intentan alcanzar un acuerdo mutuamente aceptable en la división de recursos escasos (Rahwan et al., 2003a)**

*La negociación se puede resolver con un plan común*

- **Negociación estricta:**
  - Negociación de un plan común;
  - Ejecución estricta del plan.
- **Otros aspectos de la Negociación:**
  - Existen esquemas conocidos de negociación y relajación de metas
  - Existen mecanismos de arbitraje bien conocidos

# Negociación

- **Objetivo:**
  - determinar (las condiciones de) un acuerdo entre, al menos, dos agentes
- **Ejemplos de negociación**
  - **Red de Contratos**
    - Adjudicar productos y tareas a través de un “mercado”
    - $n$  participantes, transacción entre 2
  - **Regateo**
    - Llegar a un acuerdo entre *todos* los participantes
  - **Argumentación**
    - Resolver (supuestos) conflictos a través del debate



# Subastas

- Las subastas suelen establecer un contrato entre dos agentes (el subastador y el que gana la subasta).
- Mecanismo **estructurado** para forjar acuerdos
  - Protocolo: semi-distribuido, con diferentes roles
    - 1 subastador
    - $N$  subasteros
  - Estrategias:
    - “pujas” de los subasteros

# Elementos de un Protocolo de Subasta

- Tipo de ofertas:
  - abierto (*open-cry*): los subasteros conocen mutuamente sus ofertas
  - privado/cerrado (*sealed-bid*): los subasteros sólo conocen sus propias ofertas
- Proceso de ofertas:
  - una vuelta (*one-shot*): los subasteros sólo dan una oferta
  - directa (*forward*): el precio de las ofertas va ascendiendo
  - inversa (*reverse*): el precio de las ofertas va descendiendo
- Proceso de adjudicación:
  - ¿Qué oferta se usa para determinar el precio que ha de pagar el ganador? (*first-price, second-price, ...*)

# Subasta

Aparece en numerosas aplicaciones en informática.

Oferta 1  
Agente 3



Oferta 2  
Agente 1



Oferta 3  
Agente 3



Precio inicial

Precio privado

Oferta ganadora

## Protocolos comunes:

Subasta inglesa

Subasta holandesa:

Subasta sellada:

Subasta de Vickrey

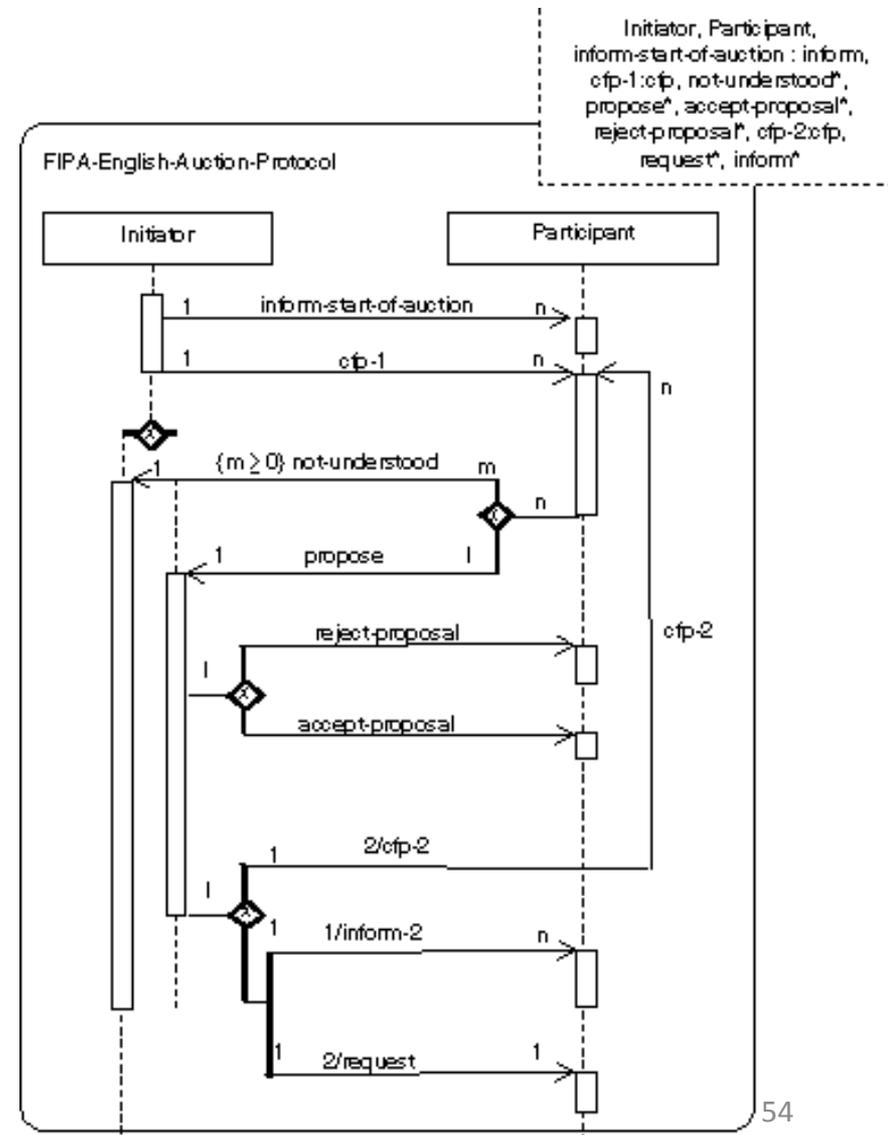
...

# Subasta inglesa

- Inicio:
  - el subastador ofrece un producto a un **precio inicial** (usualmente por debajo de un precio mínimo privado)
- Apuestas:
  - los subasteros van ofertando precios (ninguna, una, o varias veces)
  - cada oferta tiene que **superar** todas las anteriores
  - el ciclo de apuestas termina cuando no hay más ofertas
- Adjudicación:
  - si la última oferta alcanza el precio mínimo (privado) del subastador, el producto es adjudicado al subastero de la oferta *más alta*
  - de lo contrario no se vende el producto (**el subastador tiene la última palabra!!!**)

# Subasta inglesa entre agentes: FIPA

- En la subasta de agentes, los participantes *no* están físicamente presentes en una sala de subasta
- *reject-proposal*: pueden llegar pujas ilegales, p.ej. por retardos en la red
- *cfp*: anunciar cada nueva ronda de pujas con el precio actual
- *inform*: informar a todos los participantes sobre el resultado de la subasta
- *request*: requerir que el ganador realice la transacción



# Comunicación en Agentes

- La comunicación entre agentes **permite la interacción e interoperación entre agentes**, es decir sincronizar acciones, enviar y recibir conocimiento, resolver conflictos en la resolución de una tarea, etc.
- La comunicación **permite a los agentes coordinar acciones y comportamientos** para dar lugar a sistemas coherentes, capaces de conseguir las metas propias de los agentes o globales del sistema
- Los agentes que cooperan en la realización de tareas tienen un **comportamiento social**
  - Los agentes **intercambian conocimiento**: además de datos, un mensaje tiene asociado un contenido semántico
  - Los agentes **realizan “actos de comunicación”**: Un agente influye en el conocimiento y acciones de otros agentes a través de un acto de comunicación

# Comunicación de Agentes

- Específicamente, para la interoperabilidad de los agentes es esencial:
  - **Lenguaje para definir los actos comunicativos a usar.**
    - Enfoque Procedimental
      - Son lenguajes sencillos y eficientes no aptos para esto
    - Enfoque Declarativo (por ej. ACL)
      - Intercambio de información declarativa
      - Problema de la expresividad
  - **Lenguajes de contenido** para expresar el formato de intercambio del conocimiento.
  - **Ontologías** para ponerse de acuerdo en un vocabulario y sentido común para describir un dominio.
  - **Protocolos de interacción** predefinidos para definir cual es la secuencia de mensajes esperada => conversación
    - Red de Contratos
    - Mecanismo del Mercado
    - Pizarrón

# ACTOS DE HABLA

*DEFINEN EL CONJUNTO DE ACCIONES  
INTENCIONALES EFECTUADAS EN EL  
TRANSCURSO DE UNA COMUNICACIÓN*

*PRAGMATISMO DEL DISCURSO*

# TIPOS DE ACTOS DE HABLA

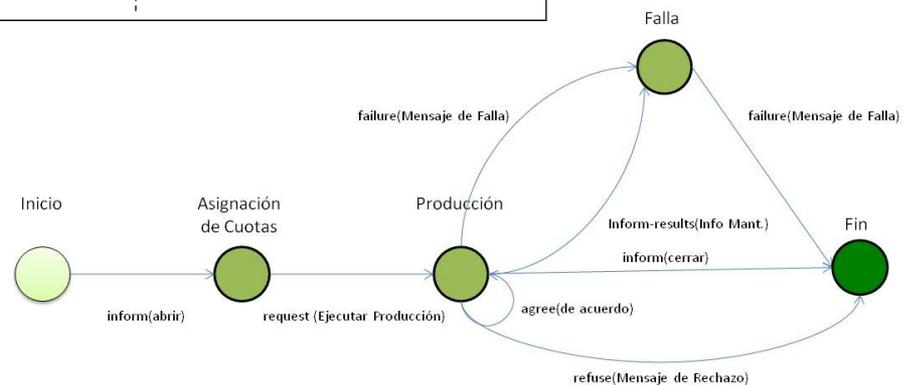
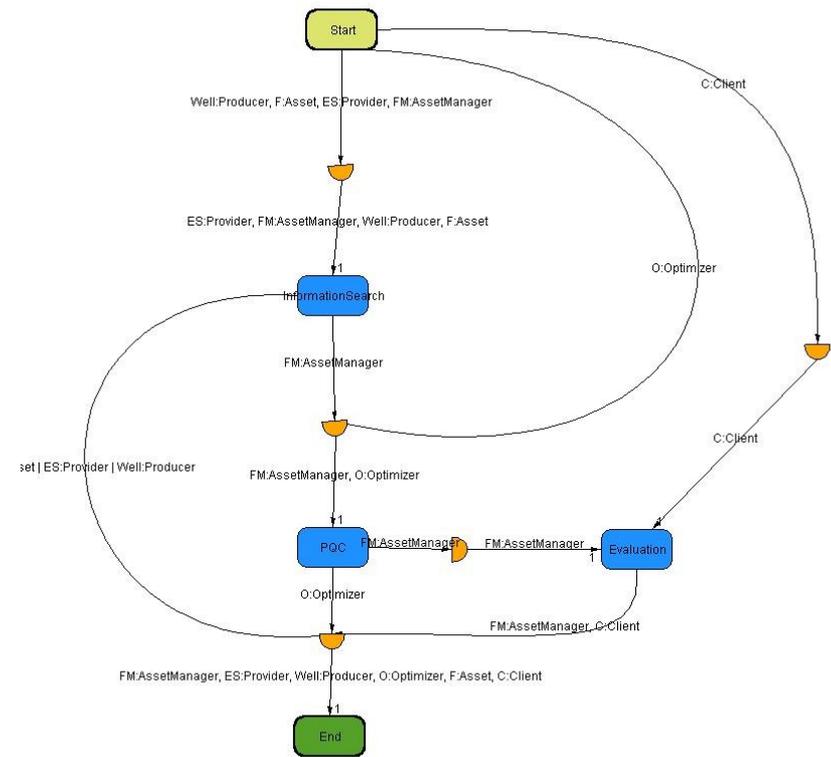
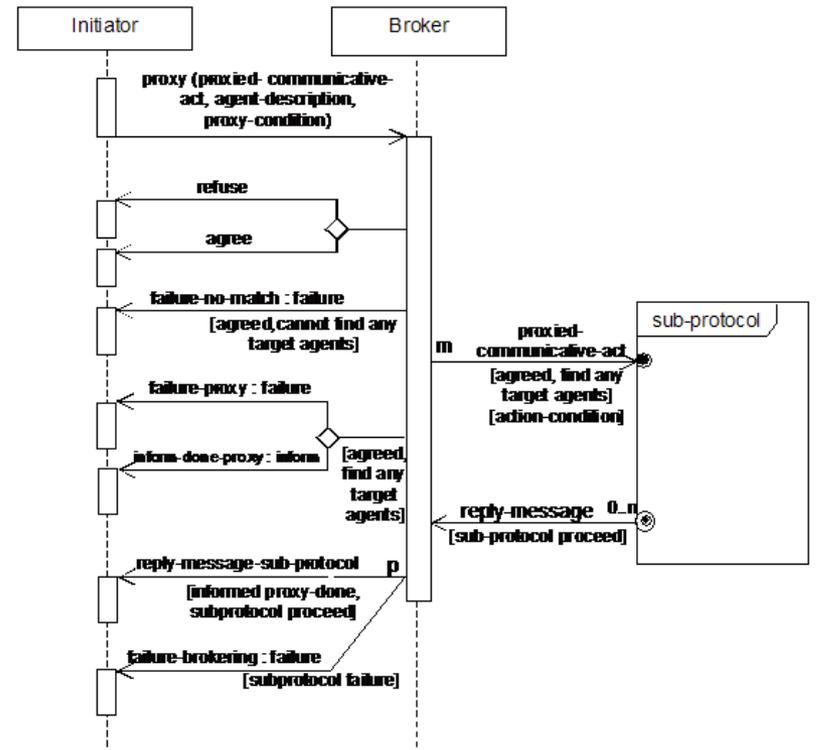
<u>TIPO</u>	<u>EJEMPLO</u>	<u>CATEGORIA</u>
DEMANDA	DEMANDA-HACER	ORDENAR
INTERROGACION	PREGUNTAR	INTERROGACION
AFIRMACION	AFIRMAR	ASEGURAR
OFERTA DE SERVICIO	OFRECER-SERVICIO	PROMETER
INDICAR HABILIDADES	SABER-HACER	EXPRESIVO
PROPOSICION HIPOTESIS	PROPONER-IDEA	EXPRESIVO

# CONVERSACIONES

- Patrón para el intercambio de mensajes entre dos o más agentes que acuerdan comunicarse entre si
- Protocolos de conversación: los agentes no realizan un simple intercambio de mensajes aislados, sino que mantienen conversaciones (Se espera cierto tipo de respuesta del interlocutor)
- Ventajas asociadas a los protocolos de conversación
  - Proporcionan un marco para el intercambio de mensajes (interpretación).
  - Posibilitan mejor ajuste a modelos intuitivos de interacción entre agentes.
  - La estructura conversacional es separable del resto del comportamiento del agente (favorece la reutilización)

# CONVERSACIONES

FIPA-Brokering-Protocol



# ACL: Agent Communication Language

- Lenguaje para intercambio de mensajes entre agentes.
- Componentes de base
  - Vocabulario
  - Lenguaje “interno”: KIF
  - Lenguaje “externo”: KQML
- Un mensaje en ACL es una expresión **KQML**, en la que los argumentos son términos o sentencias **KIF**, formadas por palabras del **vocabulario** del ACL

# ACL: Agent Communication Language

- El mensaje ACL posee los campos:
  - **sender:** agente que envía el mensaje.
  - **receivers:** lista de receptores.
  - **performative:** tipo de mensaje. Indica la intención que el emisor intenta lograr enviando el mensaje.
  - **content:** contenido principal del mensaje.
  - **language:** lenguaje usado en el contenido. P.ej. la sintaxis utilizada para expresar el contenido.
  - **ontology:** ontología usada en el contenido. P.ej. el vocabulario de símbolos usados en el contenido y su significado.
  - **conversation-id, reply-with, in-replay-to, reply-by:** para el control de conversaciones concurrentes..

# KQML

- Es un lenguaje y protocolo basado en mensajes para comunicación entre agentes.
- Los mensajes KQML no sólo comunican frases en algún lenguaje sino que mas bien comunican una *actitud* sobre el contenido
- Las primitivas del lenguaje reciben el nombre de “*performatives*”
- Así, cada mensaje posee
  - Performativa
  - Parámetros

# KQML

- **Tres capas**
  - Contenido (KIF)
  - Mensaje (Performativas)
  - Comunicación
- **Performativas**
  - De discurso: para intercambio de información y conocimiento (ask, tell)
  - De intervención y conversación: para intervenir en una conversación (sorry, next, ready)
  - De red: no son estrictamente actos de habla, pero permiten a los agentes encontrar otros agentes (recruit, register)

# KQML

- Mensajes en KQML

```
( Performativa
```

```
    : content (<Contenido a transm.>)
    : receiver <destino>
    : language <lenguaje del contenido>
    : ontology <ontologia usada>
```

```
)
```

- Ejemplo

```
( ask-one
```

```
    :content (PRICE IBM ?price)
    :receiver -stock-server
    :language LPROLOG
    :ontology NYSE-TICKS
```

```
)
```

# KQML (Realizativas o performativas)

- Se usan para los actos comunicativos (Actos de Habla)

## **Actos Comunicativos**

## **Operaciones**

para preguntas básicas

evaluate, ask-if, ask-in, ask-one, ask-all

de preguntas con varias resp.

stream-in, stream-all

de respuesta

reply, sorry

de información genérica

tell, achieve, cancel, untell ...

para control de flujo

standby, ready, next, rest, discard ...

que define capacidades

advertise, subscriber, monitor, export ...

de gestión

register, forward, broadcast, router ...

# KIF

- Originalmente desarrollado con la intención de ser un lenguaje común para expresar propiedades de un dominio en particular.
- Lenguaje de predicados de primer orden
- Notación prefija (LISP)
- Ejemplo
  - (= (temperature ml) (scalar 83 Celsius))
  - (defrelation bachelor (?x) :=
    - (and (man ?x)
    - (not (married ?x))))

# Comunicación entre agentes

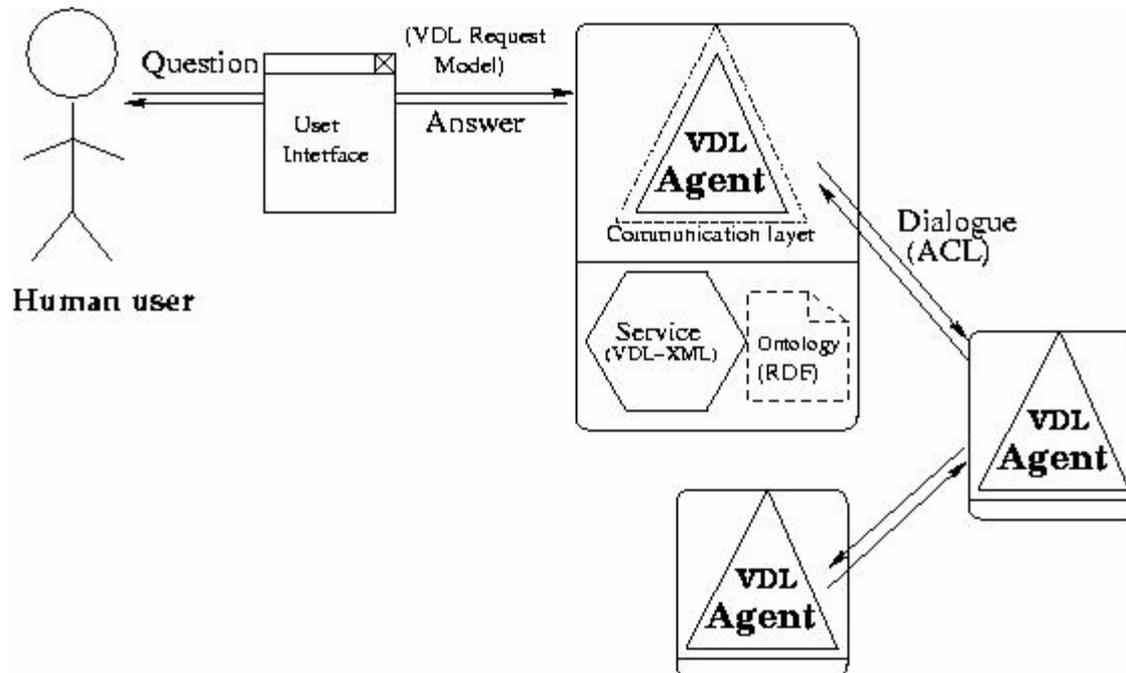
## Resumen



# Agentes Inteligentes en Aml

## Agente :

- Mecanismos de razonamiento sobre los servicios ofrecidos
- Representación del servicio
- Ejecución de servicios



# Comunidad de Agentes Inteligentes en Aml

## **Agentes del módulo de información**

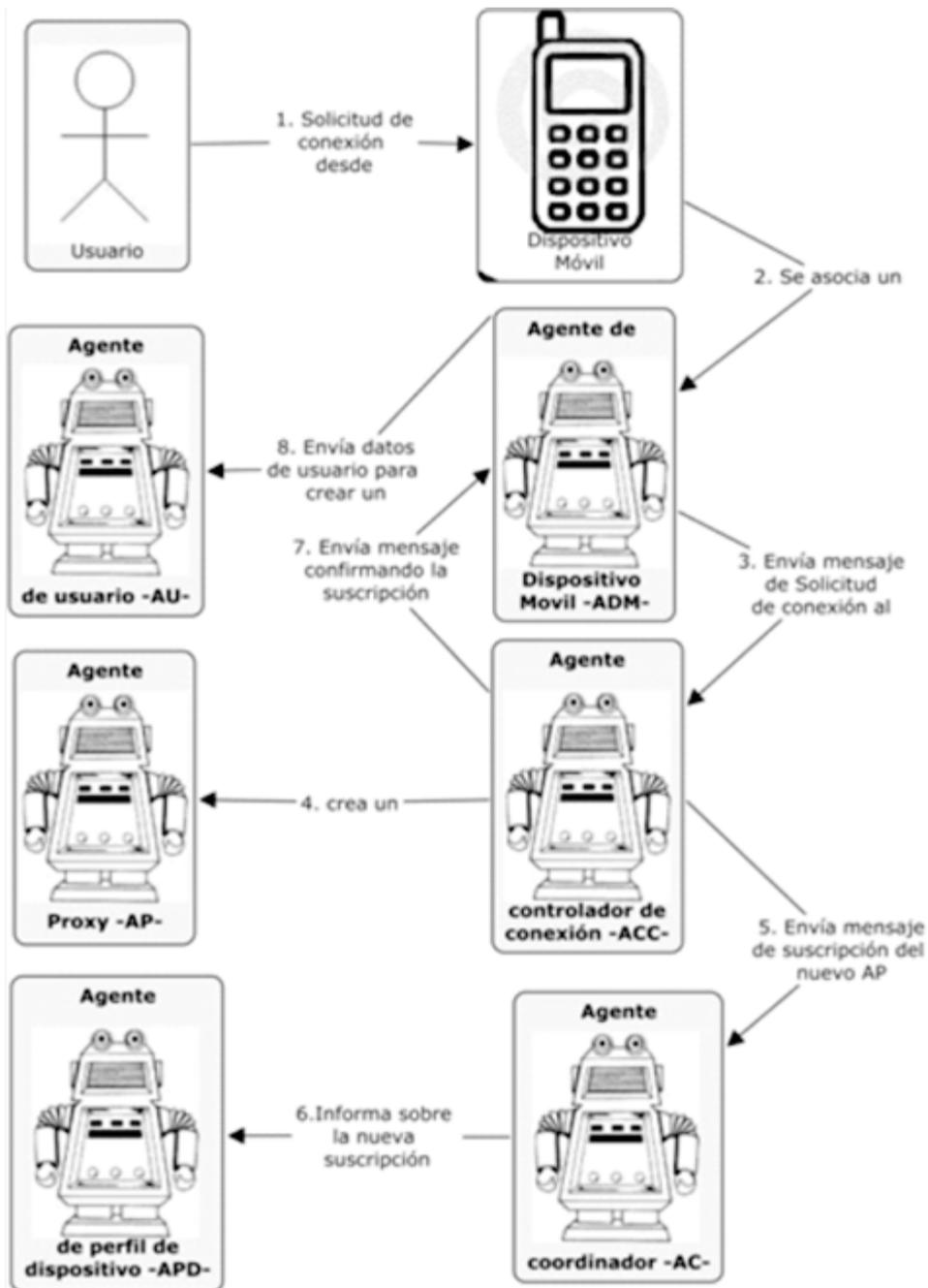
- Agente de dispositivo móvil –ADM–.
- Agente de perfil de dispositivo –APD–.
- Agente de usuario –AU–.

## **Agentes del módulo de comunicación**

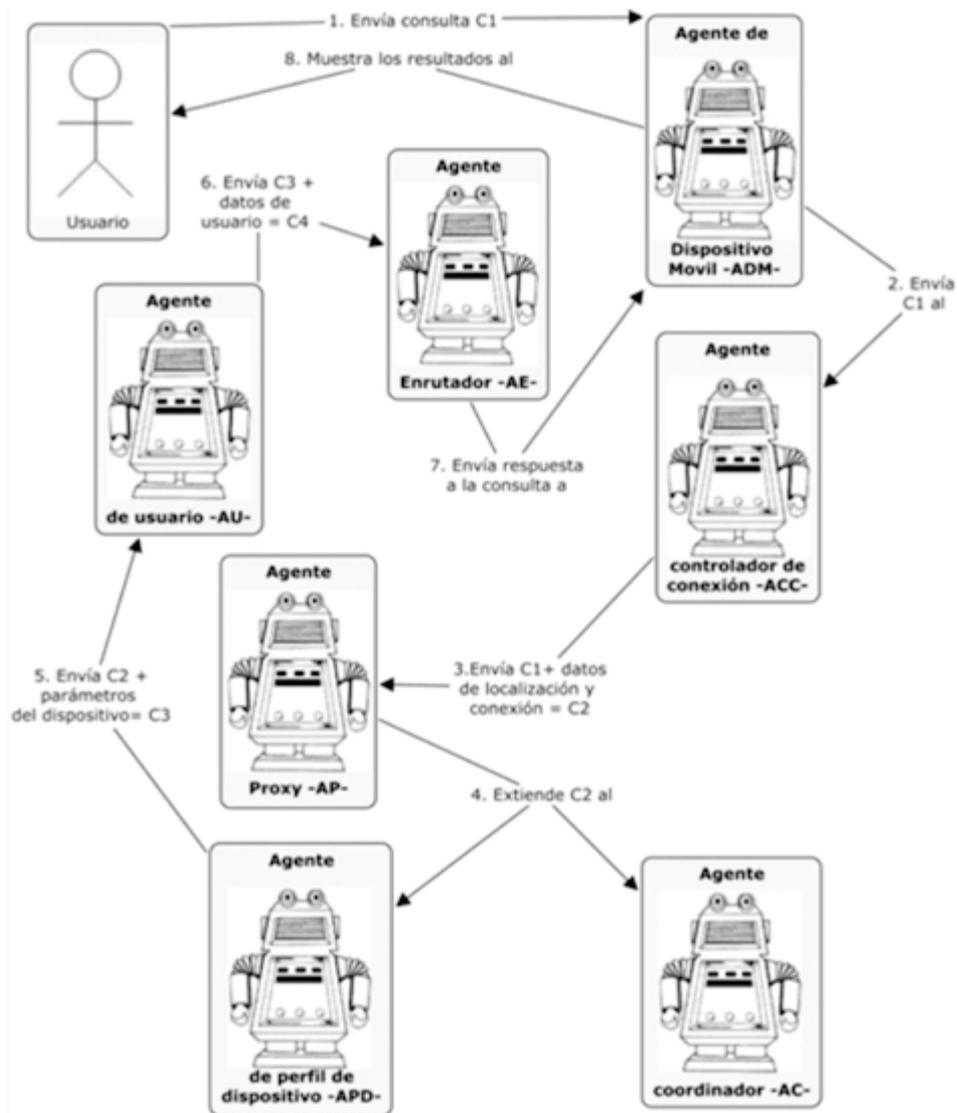
- Agente controlador de conexión –ACC–.
- Agente coordinador –AC–.
- Agente proxy –AP–.

## **Agentes del módulo de adaptación**

- Agente enrutador –AE–.



**Flujo de mensajes para el escenario de conexión**



**Paso de mensajes para la solicitud de consulta del usuario**



**FIPA**

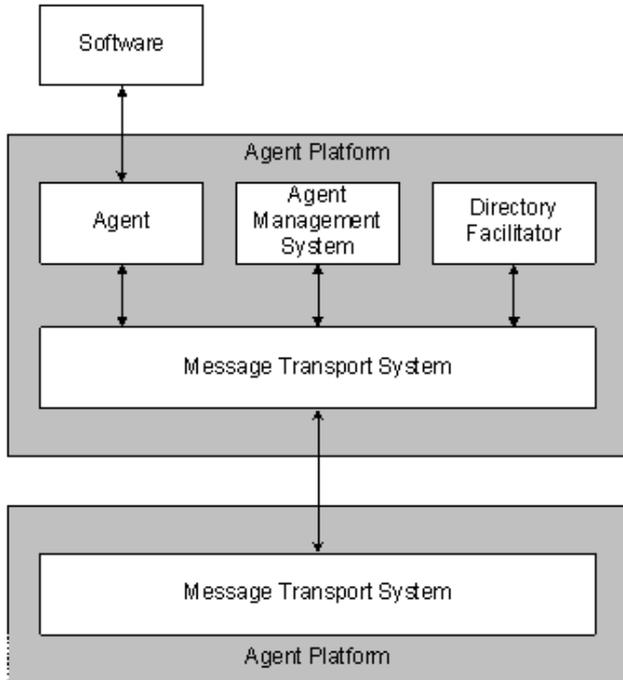
- La necesidad de tener estándares y normas en los SMA viene dado por las siguientes razones:
  - Favorece el desarrollo de productos de software basados en el paradigma de agentes.
  - Coadyuva a la definición de teorías, modelos, etc., alrededor de la teoría de agentes.
  - Promueve las características deseadas de las aplicaciones basadas en agentes: abiertas, para ambientes heterogéneos y con propiedades emergentes.
- FIPA busca establecer o garantizar, entre otras cosas:
  - La definición de una arquitectura genérica para soportar SMA.
  - La interoperabilidad del transporte de mensajes entre agentes.
  - Las diversas formas del lenguaje de contenidos de los agentes, a ser usados en sus comunicaciones (*ACL*).

# Agentes: FIPA

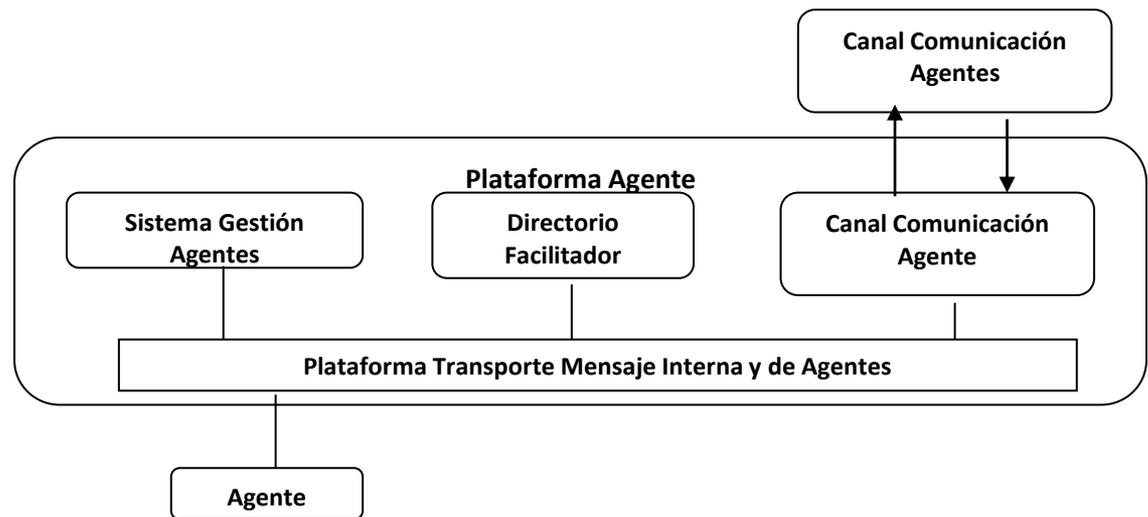
- En 2002 se estableció la Fundación para los Agentes Físicos Inteligentes (FIPA), la cual es una organización de la Sociedad de Computación de la IEEE que promueve la tecnología basada en agentes y la interoperabilidad de los agentes desarrollados con diferentes tecnologías .
- La FIPA ha desarrollado una serie de especificaciones, las cuales son una colección de estándares:
  - marco arquitectónico genérico para el despliegue de SMAs,
  - protocolos de comunicación,
  - Lenguajes de contenido,
  - mecanismos de gestión e implantación de SMAs,
  - estándares de calidad de servicio,
  - aplicaciones comunes basadas en agentes , entre otras.

# Agentes: FIPA

- Arquitectura Abstracta
  - FIPA Abstract Architecture Specification
  - FIPA Domains and Policies Specification
- Actos comunicativos
  - FIPA Communicative Act Library Specification
- Lenguajes de contenido
  - FIPA SL Content Language Specification
  - FIPA CCL Content Language Specification
  - FIPA KIF Content Language Specification
  - FIPA RDF Content Language Specification
- Gestión de agentes
  - FIPA Agent Management Specification
  - FIPA Agent Discovery Service Specification
  - FIPA JXTA Discovery Middleware Specification



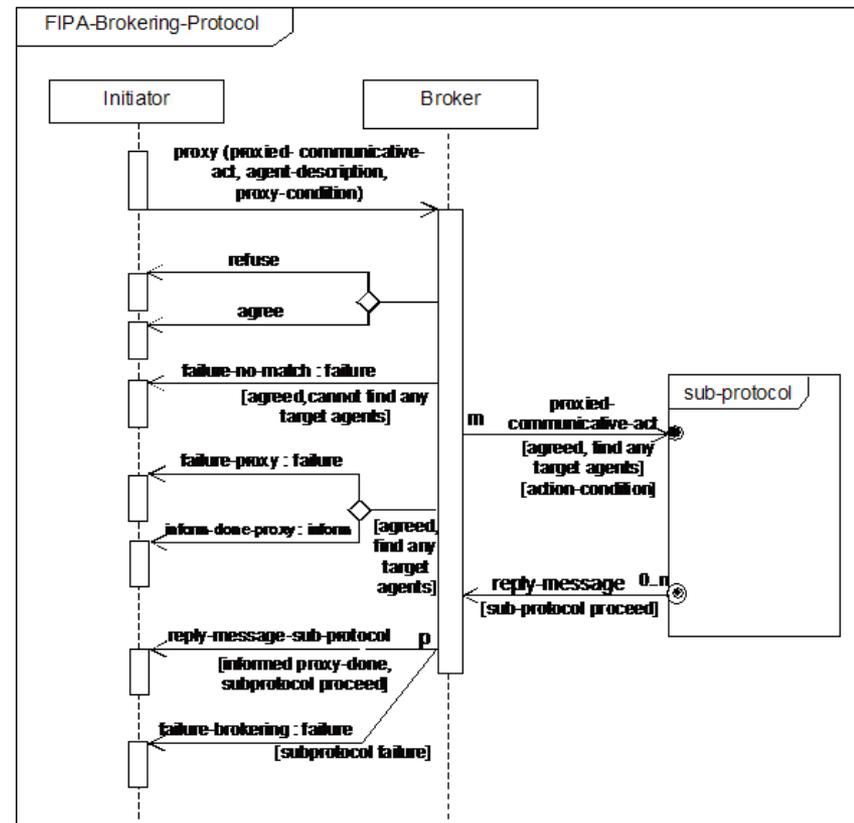
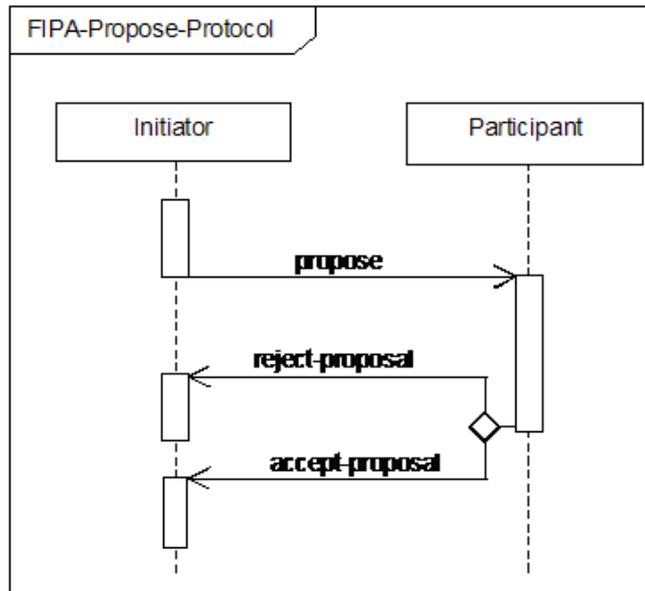
Estándar de plataforma  
de servicios para el  
despliegue de SMAs



# Agentes: FIPA

- Lenguajes de comunicación
  - Protocolos de interacción
    - FIPA Request Interaction Protocol Specification
    - FIPA Query Interaction Protocol Specification
    - FIPA Request When Interaction Protocol Specification
    - FIPA Contract Net Interaction Protocol Specification
    - ...
    - FIPA Brokering Interaction Protocol Specification
    - FIPA Recruiting Interaction Protocol Specification
    - FIPA Subscribe Interaction Protocol Specification
    - FIPA Propose Interaction Protocol Specification
    - FIPA English Auction Interaction Protocol Specification
    - FIPA Dutch Auction Interaction Protocol Specification

# Ejemplos de Protocolos de interacción



# Agentes: FIPA

- Transporte de mensajes
  - Representaciones de ACL
    - FIPA ACL Message Representation in Bit-Efficient Specification
    - FIPA ACL Message Representation in String Specification
    - FIPA ACL Message Representation in XML Specification
  - Representaciones de envoltorios
    - FIPA Agent Message Transport Envelope Representation in XML Specification
    - FIPA Agent Message Transport Envelope Representation in Bit Efficient Specification
  - Protocolos de transporte
    - FIPA Agent Message Transport Protocol for IIOP Specification
    - FIPA Agent Message Transport Protocol for HTTP Specification
    - FIPA Agent Message Transport Protocol for WAP Specification

# Agentes: FIPA

- Aplicaciones:
  - FIPA Nomadic Application Support Specification
  - FIPA Quality of Service Specification
  - FIPA Personal Travel Assistance Specification
  - FIPA Audio-Visual Entertainment and Broadcasting Specification
  - FIPA Network Management and provisioning Specification
  - FIPA Personal Assistant Specification
  - FIPA Message Buffering Service Specification

## NOMBRAMIENTO DE AGENTES

- **Nombrado de agentes estático:** cada agente tiene un identificador único (GUID) :

***<name>@<hostname>:<port>/<target>***

- Name : identificador del agente en la plataforma de agentes a la que pertenece.
  - HostName: Dirección IP (o nombre DNS) del host en el cual se está ejecutando su ACC.
  - Port: Puerto en el que escucha el ACC.
  - Target: Identificador del agente que debería recibir el mensaje para encaminarlo al agente destino. Normalmente será el propio ACC.
- Ejemplo de GUID podría ser: `acc@iiop://www.ucm.es:50/acc`

# Agentes: FIPA

## NOMBRAMIENTO DE AGENTES

- Estructura del nombre más flexible y completa

**AgentName= (":guid" Word NameFields\*)**

**NameFields= ":addresses" (" CommAddress+ ")**

**| ":resolver" (" AgetName+ ")**

**| ":authenticator" (" AgentName+ ")**

- GUID: Identificador único del agente en la plataforma de agentes. No tiene porque contener la dirección de transporte como en el modelo simple.
- Addresses: Dirección/es de transporte en la que se encuentra el agente.
- Resolver: Dirección donde se encuentra el Servidor de Nombres (ahora en el AMS)
- Authenticator: Dirección del agente de autenticación (ahora el propio AMS).

# Agentes: FIPA

## NOMBRAMIENTO DE AGENTES

- Un ejemplo podría ser:

```
:guid MiAgente@iiop://www.ucm.es
```

```
:resolver (
```

```
:guid ams@www.ucm.es
```

```
:addresses (iiop://www.ucm.es:81/ams
```

```
http://www.ucm.es:80/ams)
```

```
)
```

```
:addresses (iiop://fipa.org/MiAgente)
```

```
:authenticator (iiop://fipa.org/Autenticación)
```



Plataformas más comunes  
para el desarrollo y/o el  
despliegue de agentes (MGS  
para SMA) y Modelos de  
Referencia SMA

# Plataformas y Modelos de Referencia

- **Plataformas más comunes para el despliegue (MGS):**
  - Ambientes híbridos basados en FIPA: JADE, JACK, Zeus, Fipa-OS, etc.
  - Otros ambientes híbridos: SEAGENT, Multi-AGent Environment – MAGE, DECAF (Distributed, Environment-Centered Agent Framework), etc.
  - Otros: MaDKit (Multiagent Development Kit), SPADE (Smart Python multi-Agent Development Environment), etc.
- **Plataformas más comunes para el desarrollo:**
  - Ambientes híbridos basados en FIPA: JADE, JACK, Zeus, Fipa-OS, etc.
  - Otros ambientes híbridos: SEAGENT, Multi-AGent Environment – MAGE, DECAF (Distributed, Environment-Centered Agent Framework), etc.
  - Otros ambientes: Amine, AgentTool, AgentBuilder
- **Modelos de Referencia :**
  - PABADIS (Plant Automation based on Distributed Systems),
  - AARIA (Autonomous Agents for Rock Island Arsenal),

# Plataformas y Modelos de Referencia ULA

- **Plataformas más comunes para el despliegue (MGS):**
  - Medio de Gestión de Servicios orientado a Sistemas Tiempo Real.
- **Plataformas más comunes para el desarrollo:**
  - EDISMA: Entorno de Desarrollo Integrado para la construcción de SMA
- **Modelos de Referencia:**
  - Sistema Automatizado Distribuido Inteligente basado en Agentes (SADIA)
  - Sistema de Control Distribuido Inteligente basado en Agentes (SCDIA)
- **Libro**
  - “Sistemas MultiAgentes y sus Aplicaciones en Automatización Industrial”, J. Aguilar, A. Ríos, F. Hidrobo, M. Cerrada, 2da Edición, Universidad de Los Andes,, 2013..

## JADE (Java Agent DEvelopment Framework)

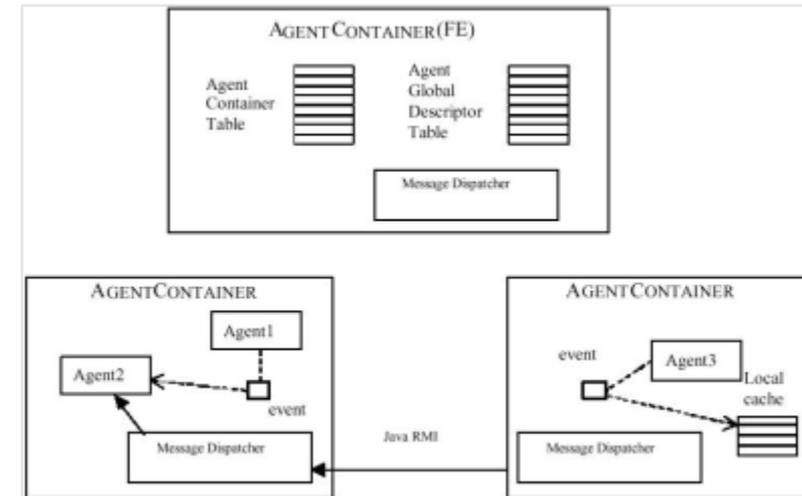
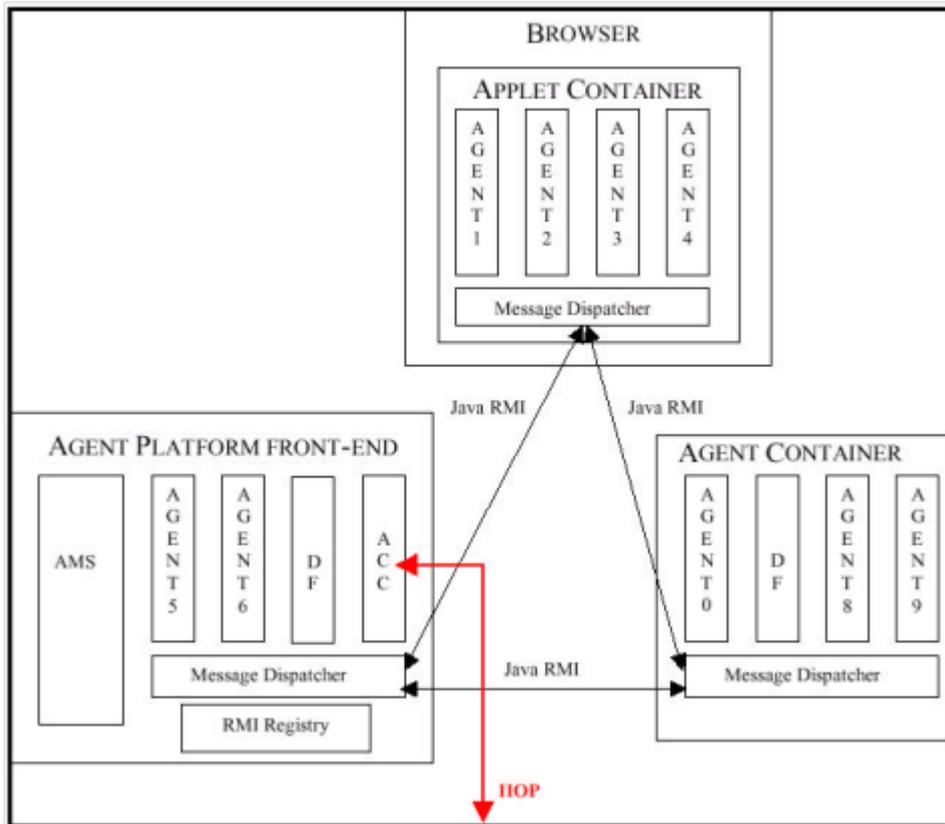
- El objetivo de JADE es simplificar el desarrollo de agentes, garantizando seguir los estándares establecidos por la FIPA, a través de un conjunto de servicios ofrecidos por la plataforma.
- JADE puede ser considerado como un *middleware de agentes*, que implementa una plataforma para la gestión de agentes y un IDE (*Integrated Development Environment*) para el desarrollo de agentes.
- En particular, incluye:
  - Una plataforma de agentes compatible con FIPA.
  - Un paquete basado en Java para desarrollar agentes.
  - Varias extensiones para ejecutarse en dispositivos con aplicaciones específicas (PDA, teléfonos inteligentes, etc.).

# JADE

- JADE ha sido totalmente implementado en el lenguaje Java.
- Extensiones recientes de JADE son compatibles con los entornos móviles de Java (J2ME CLDC MIDP 1.0).
- JADE es un software libre distribuido por Telecom Italia, bajo los términos de la LGPL (*Lesser General Public License*) .
- La plataforma de agentes se puede distribuir en varios *hosts*. *Es suficiente* una aplicación Java, y por lo tanto una Máquina Virtual Java (JVM) se ejecuta en cada host.
  - Cada JVM es básicamente un contenedor de agentes que proporciona un completo entorno de ejecución para éstos, y permite que múltiples agentes se ejecuten simultáneamente en el mismo host.

# JADE

## En cuanto al contenedor

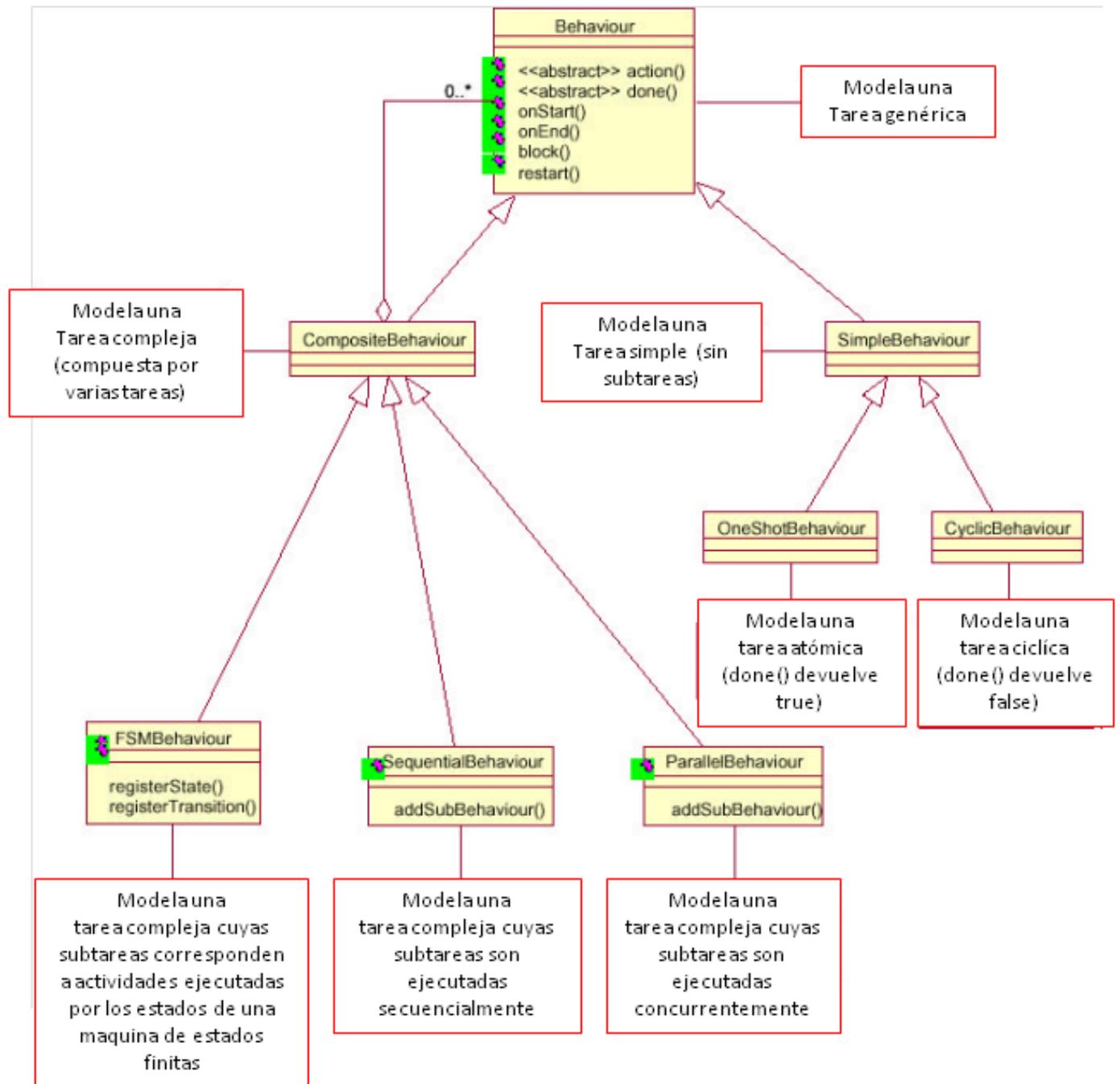


El contenedor FE es el que provee todos los servicios FIPA para darle soporte a la plataforma de agentes provista por JADE

# JADE

- Para la implantación de SMA, JADE proporciona:
  - Un entorno de ejecución en el que los agentes se ejecutan.
  - Bibliotecas de clases para la creación de agentes mediante la herencia y la redefinición de comportamientos.
  - Un conjunto de herramientas gráficas para el monitoreo y la administración de la plataforma.
- JADE define la Clase *Agent*, la cual es una superclase que permite a los usuarios crear agentes JADE. Esta clase suministra métodos que permiten ejecutar las tareas básicas de los agentes como:
  - Pasar mensajes utilizando objetos ACLMessage
  - Dar soporte al ciclo de vida de un agente.
  - Planificar y ejecutar múltiples actividades al mismo tiempo.

# Clases de JADE

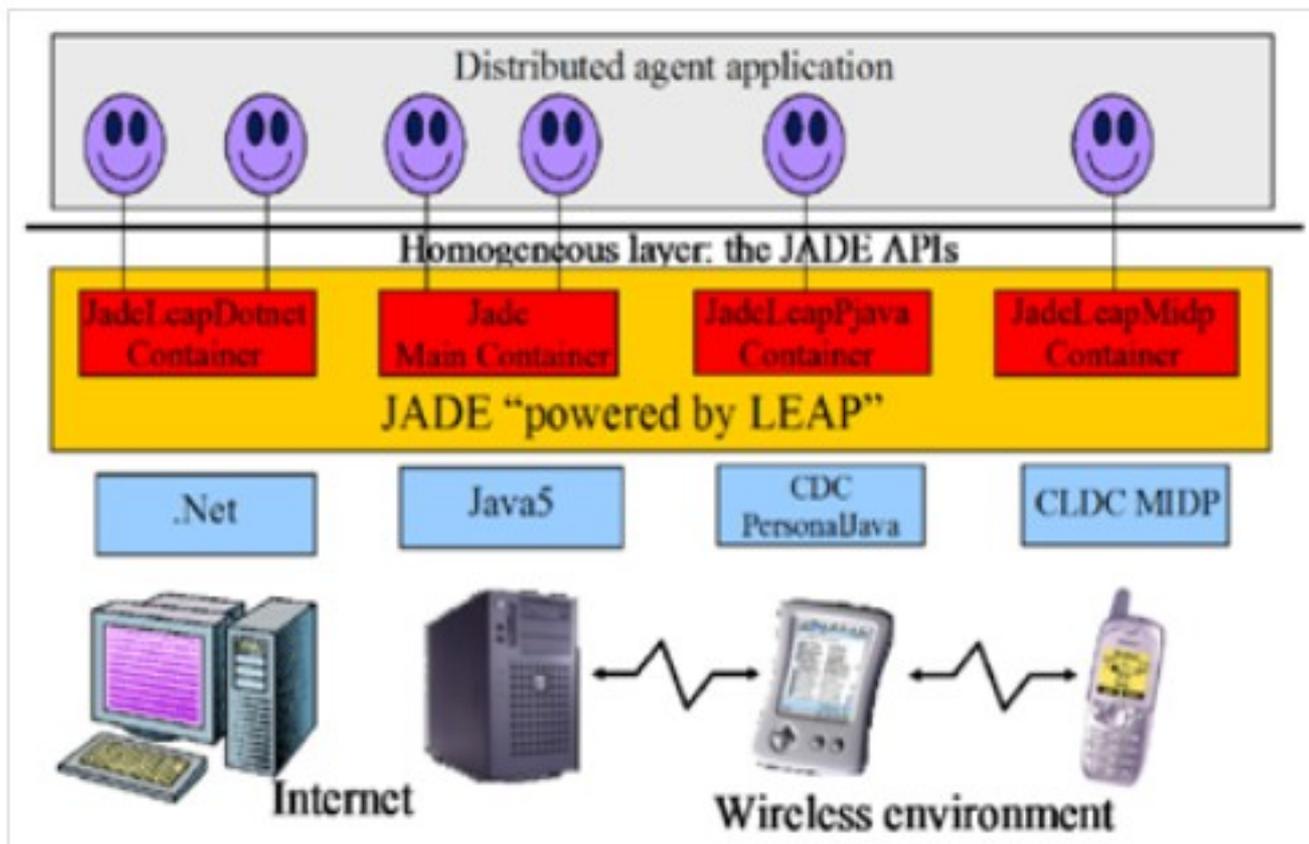


# JADE

- El “comportamiento” de un agente define las acciones a realizar bajo un determinado evento.
  - Los diferentes comportamientos que el agente adopta se definen a partir de la clase *Behaviour*, para lo cual usa el método *addBehaviour*.
  - La clase *Behaviour* contiene los métodos: i) *action()*: Se ejecuta cuando la acción tiene lugar; ii) *done()*: Se ejecuta al finalizar el comportamiento.
- Otros métodos de la Clase *Agent*: *onStart()*, *onEnd()*, *block()*, *restart()*, *blockingReceive()* y *receive()*,
- JADE ofrece una llamada a *JessBehaviour* que permite la integración con *JESS*, un entorno para la programación de reglas con su motor de razonamiento,
- JADE tiene una extensión, *WADE (Workflows and Agents Development Environment)*, sistema de workflow que permite crear procesos mediante un editor gráfico llamado *WOLF*

# JADE-LEAP

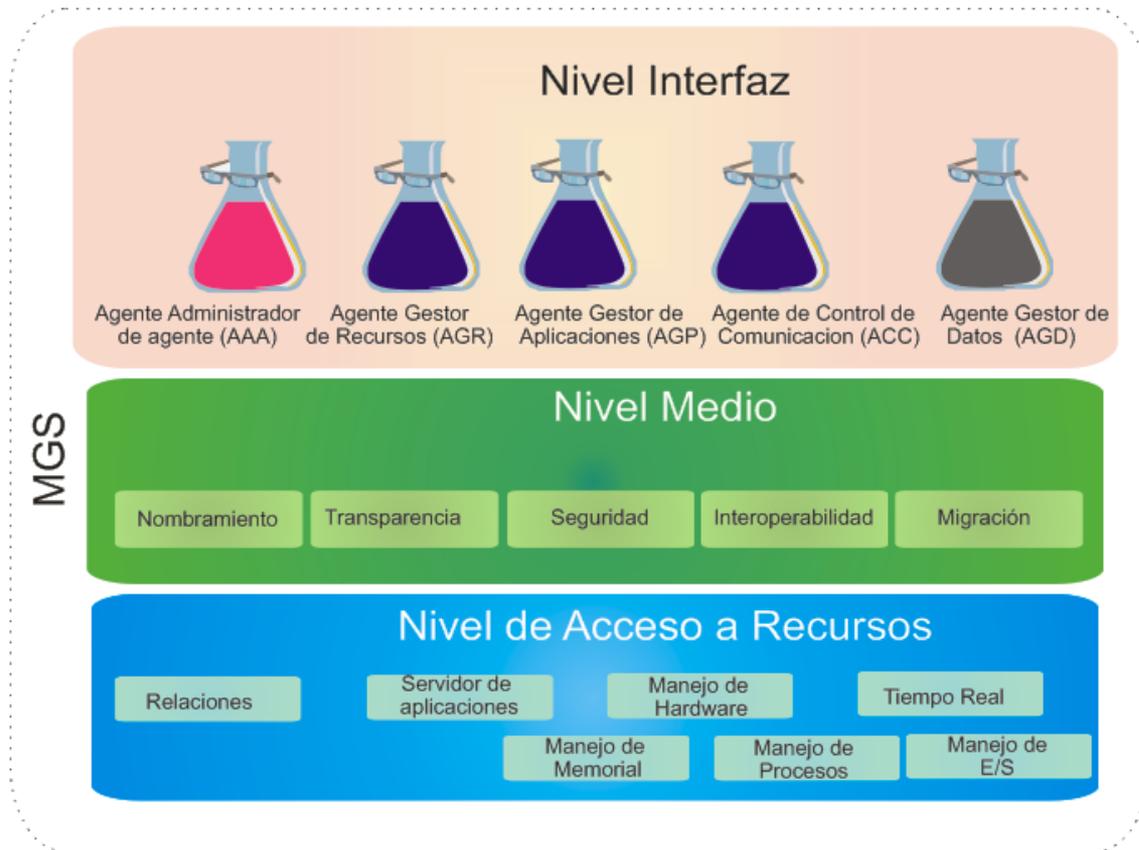
Nombre de un proyecto de la *Comunidad Económica Europea*, que ha desarrollado la biblioteca LEAP, la cual puede ser usada desde la plataforma JADE, proporcionando un entorno de ejecución para aplicaciones móviles basada en Java (J2ME CLDC).



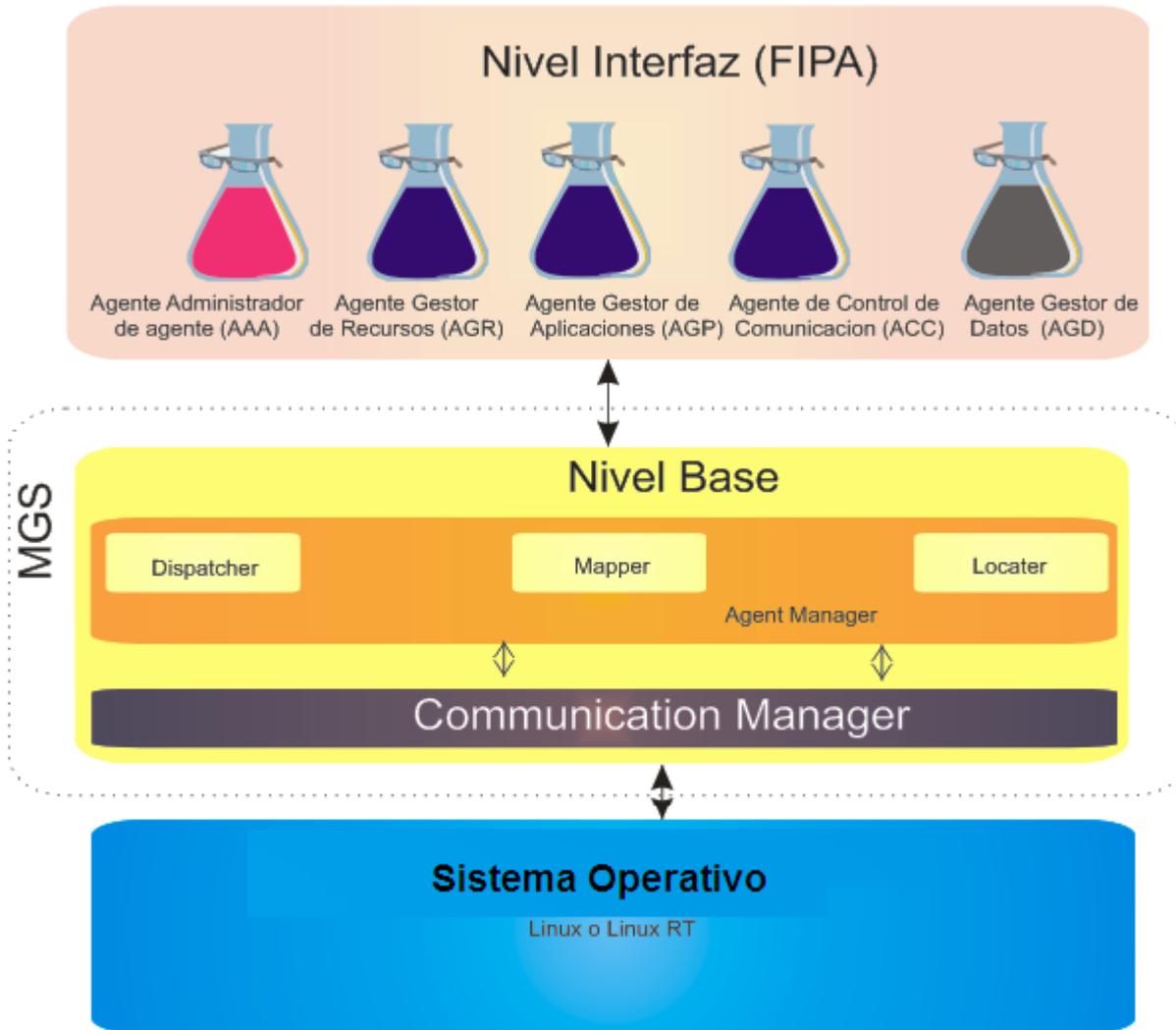
# Medio de Gestión de Servicios orientado a Sistemas Tiempo Real.



Sistema Multiagentes



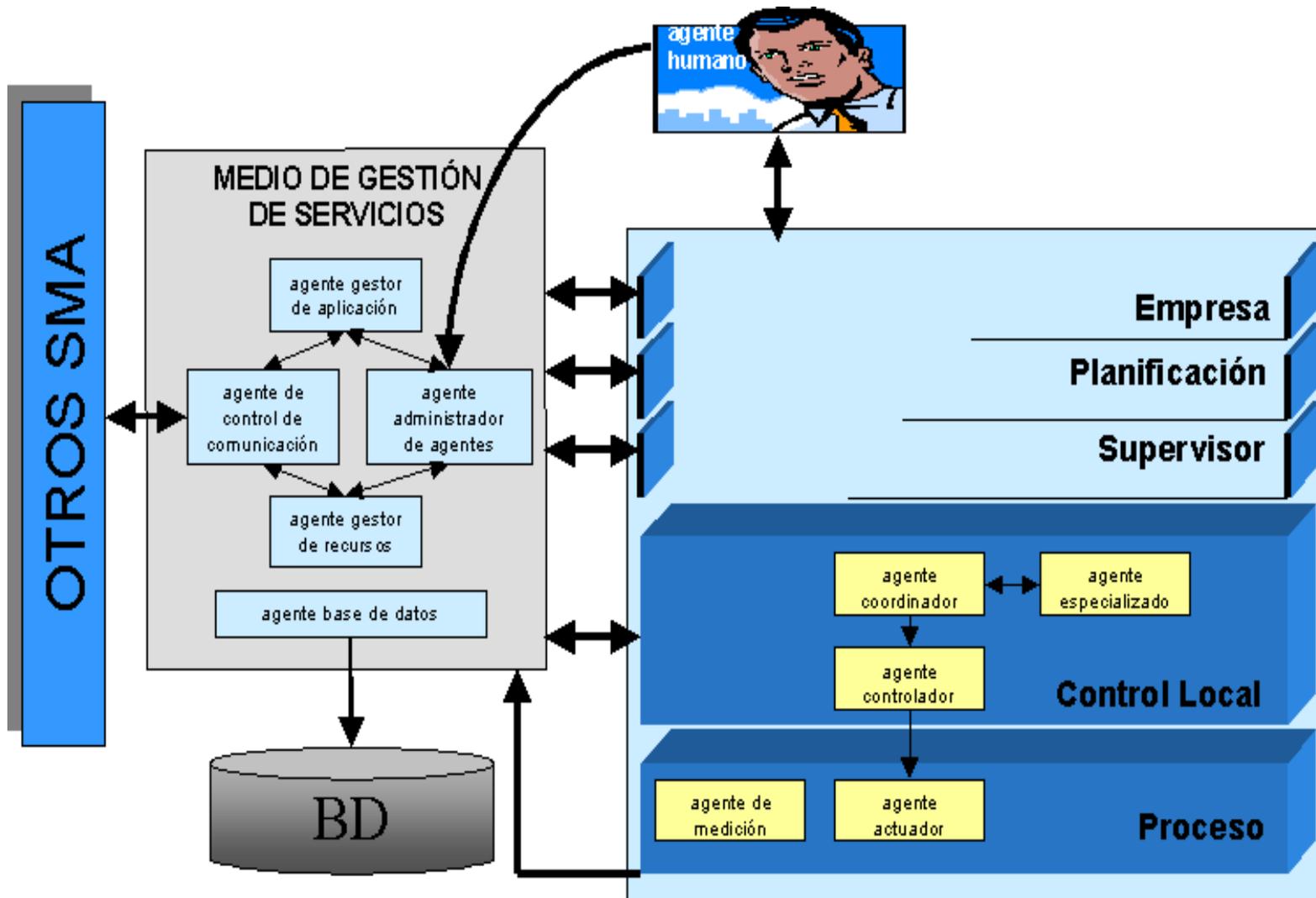
# Medio de Gestión de Servicios orientado a Sistemas Tiempo Real.



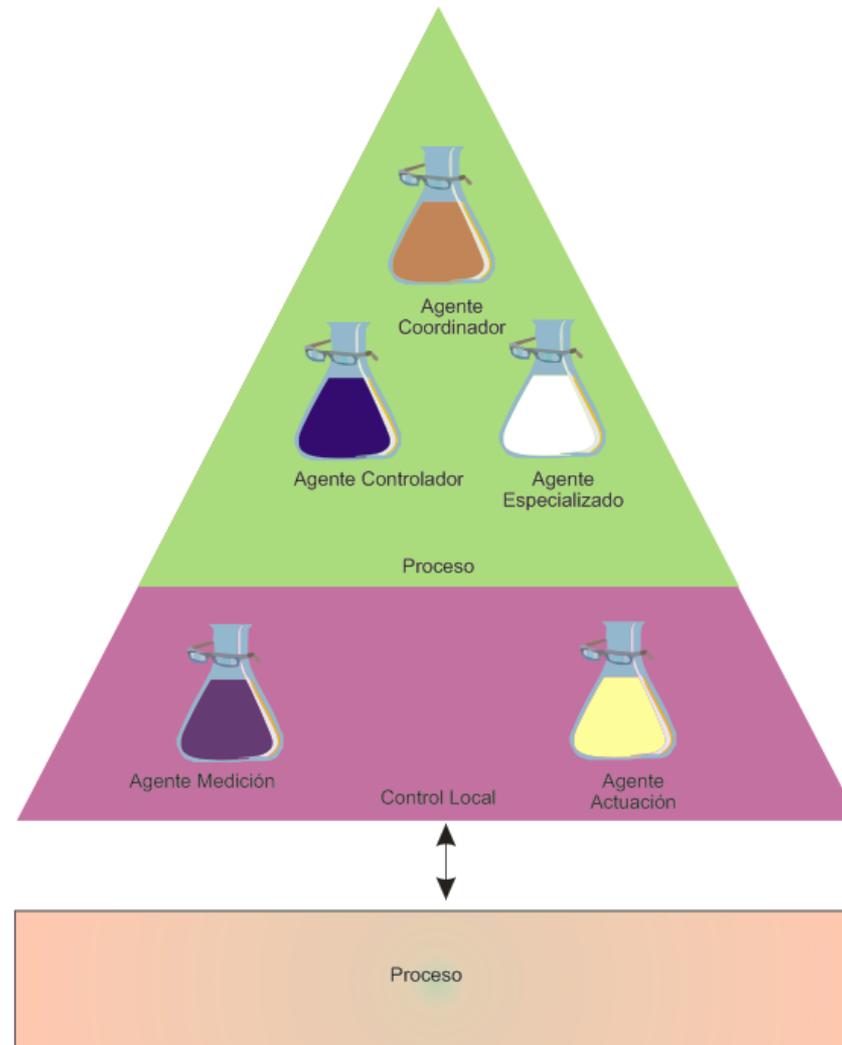
# SCDIA

## Sistemas de Control Distribuido Inteligentes Basados en Agentes

- **Características:**
  - Sistema descentralizado más que centralizado,
  - Emergente más que planificado,
  - Autónomo más que controlado,
  - Concurrente más que secuencial.
- **Habilidades**
  - ➤ Aprendizaje
  - ➤ Búsqueda
  - ➤ Inferencia y predicción.
  - ➤ Representación de fuentes de conocimiento
  - ➤ Evolutivas.



# SCDIA



# Tareas del SCDIA

## Tareas de Medición

- Medir – 1
  - Sensar – 1.1
  - Acondicionar – 1.2
- Procesar variable – 2

## Tareas de Control

- Toma de decisiones – 1
- Obtención acción de control – 2
- Procesar acción de control – 3
- Ejecutar acción de control – 4

## Tareas de especializado

- Procesar agente especializado - 1

## Tareas de manejo de información

- Almacenar información –1
- Actualizar información - 2
- Procesar (del A. BD) - 3
- Evaluar (del A. BD) - 4
- Buscar información - 5
- Gestión BD – 6

## Tareas de localización

- Localizar recurso
- Localizar agente especializado de recurso
- Localizar aplicación
- Loc. Ag. Especializado en aplicación
- Asignación de tarea/recurso
  - optimización

## Tareas de planificación

- Procesar Agente coordinador – 1
  - Evaluar
  - Tomar decisiones
  - Cambiar atributos de agentes

## Tareas de activación de agentes

- Inicializar agente – 1
  - Asociar nombre, nodo y estado – 1.1
  - Comunicar agente de recurso – 1.2
- Migrar agente – 2
- Cambiar estado – 3
- Administrar seguridad – 4
- Destruir agente – 5

# Servicios del SCDIA

- Medición de variables del proceso
- Conversión de variables
- Detección de cambios bruscos.
- Conversión de señales de control en señales físicas.
- Transmisión de información
- Definición de acción de control
- Procesamiento de datos
- Búsqueda de información
- Almacenamiento de datos
- Actualización
- Identificación de medios de transmisión de datos.
- Seguridad de datos
- Evaluación de situaciones
- Coordinación de actividades (metas)
- Planificación de tareas
- Descripción de agentes
- Localización de agentes
- Actualización de agentes
- Migración de agentes
- Seguridad a nivel agentes
- Capacidad de generar restricciones
- Ubicación de aplicaciones
- Inicialización
- Finalización de comunicación
- Envío de mensajes.

# SADIA

- Modelar la arquitectura de automatización como SMA
- 3 niveles de abstracción
  - Objetos de Negocio
  - Actividades de Automatización
  - Funciones
- Cada nivel de abstracción conforma un SMA

## Primer Nivel de Abstracción



## Segundo Nivel de Abstracción



## Tercer Nivel de Abstracción (Depende del Marco de Referencia a adoptar)



- Representa unidades de producción como agentes
- En este nivel la arquitectura es metamorfica

- El sistema de agentes capaces de realizar todas las actividades relacionadas con automatización industrial.
- Sigue el modelo funcional del descrito en el estándar del AIA 95.00.01,:
- Agentes de control
- Agentes de la supervisión
- Planeamiento y gerencia de los factores de producción
- Cada agente del primer nivel tiene una arquitectura del SMA según este nivel

- En este nivel los agentes desarrollan funciones específicas sobre el proceso productivo. Cada agente del nivel anterior se modela usando un SMA en este nivel. Cualquier marco de referencia de SMA se podría utilizar en este nivel, nosotros usamos el SCDIA Cada agente del 2do. nivel tiene una arquitectura del SMA según este nivel

Objetos de Negocio

Actividades de Automatización

Funciones

# SADIA

## Primer Nivel de Abstracción

- Modela cada unidad de producción de la UP como un agente



# SADIA

## Segundo Nivel de Abstracción

- Modela las actividades vinculadas con la automatización industrial



# SADIA

## Tercer Nivel de Abstracción

- Modela las funciones vinculadas con cada actividad del segundo nivel



**Todas las  
Arquitecturas  
de referencia  
suponen  
un MGS  
orientado a  
agentes**

