



Inteligencia Artificial e Ingeniería del Conocimiento

- 📖 I. A. Clásica
- 📖 "Nuevos" enfoques de la I. A.
 - ✓ **Agentes Inteligentes**
 - ✓ Definición
 - ✓ Tipos
 - ✓ Estructura de un agente
 - ✓ Entornos de actuación
 - ✓ Aprendizaje
 - ✓ Lógicas multivaluadas
 - ✓ Computación evolutiva

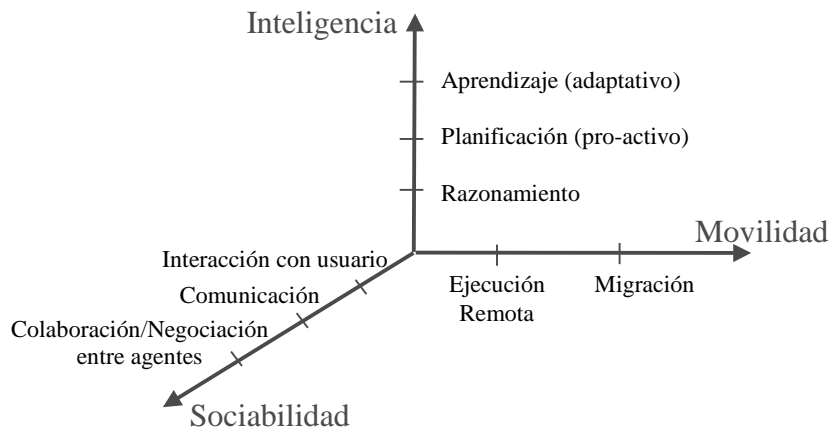


Definición I

- 📖 Un agente inteligente es todo aquello que **percibe** su ambiente mediante sensores y que **responde o actúa** de forma inteligente
 - ✓ agente=arquitectura+programa
- 📖 Conceptos relacionados
 - ✓ Softbots: Robots de software
 - ✓ Agentes móviles: tienen la capacidad de moverse por distintos nodos de una red (una o más veces)
 - ✓ perspectiva de sistema distribuido
 - ✓ Características de movilidad, persistencia, de *sociabilidad*: comunicación y colaboración



Definición (y II)



Ejemplos

Tipo de agente	Percepciones	Acciones	Metas	Ambiente
Sistema para diagnósticos médicos	Síntomas, evidencias y respuestas del paciente	Preguntas, pruebas, tratamientos	Paciente saludable, reducción al mínimo de costos	Paciente, hospital
Sistema para el análisis de imágenes de satélite	Pixels de intensidad y colores diversos	Imprimir una clasificación de escena	Clasificación correcta	Imágenes enviadas desde un satélite en órbita
Robot clasificador de partes	Pixels de intensidad variable	Recoger partes y clasificarlas poniéndolas en botes	Poner las partes en el bote que les corresponda	Banda transportadora sobre la que se encuentran las partes
Controlador de una refinería	Lecturas de temperatura y presión	Abrir y cerrar válvulas; ajustar la temperatura	Lograr pureza, rendimiento y seguridad máximos	Refinería
Asesor interactivo de inglés	Palabras escritas a máquina	Ejercicios impresos, sugerencias y correcciones	Que el estudiante obtenga la máxima calificación en una prueba	Grupo de estudiantes



Tipos I

Esqueleto de agente

Funcion ESQUELETO-AGENTE(*percepción*) return *acción*
 Static: *memoria*, la memoria del mundo del agente

```

memoria ← ACTUALIZACIÓN-MEMORIA(memoria, percepción)
acción ← ESCOGER LA MEJOR ACCIPON (memoria)
memoria ← ACTUALIZACIÓN-MEMORIA(memoria, acción)

return acción

```

Tipos de agentes

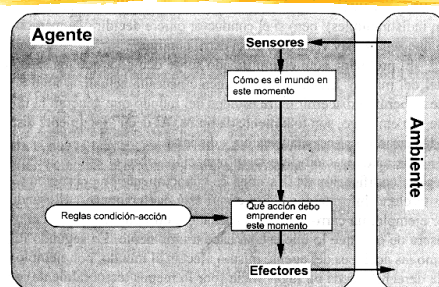
- ✓ Reflejo simple
- ✓ Bien informados (reflejo de estado interno)
- ✓ Basados en metas
- ✓ Basados en utilidad



Tipos II

Agente reflejo simple

- ✓ Reglas
- ✓ (condición --> Acción)



function SIMPLE-REFLEX-AGENT(*percept*) returns *action*

función AGENTE-REFLEJO-SIMPLE(*percepción*) responde con una *acción*

estático: *reglas*, un conjunto de reglas de condición-acción

estado ← INTERPRETAR-ENTRADA(*percepción*)

regla ← REGLA-COINCIDENCIA(*estado*, *reglas*)

acción ← REGLA-ACCIÓN[*regla*]

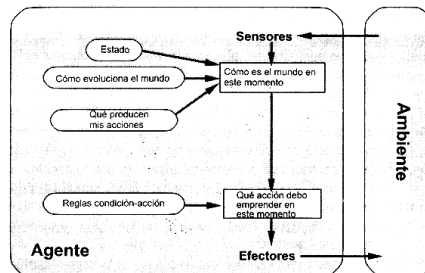
responder con una *acción*



Tipos III

Agente reflejo de estado interno

- ✓ Generación de una representación del "estado" del ambiente
- ✓ Percepciones
- ✓ Actuaciones



función REFLEX-AGENT-WITH-STATE(*percept*) **returns** *action*

función AGENTE-REFLEJO-CON-ESTADO(*percepción*) **responde** con una *acción*

estático: *estado*, una descripción prevaleciente del estado del mundo

reglas, conjunto de reglas condición-acción

estado ← ACTUALIZAR-ESTADO(*estado*, *percepción*)

regla ← REGLA-COINCIDENCIA(*estado*, *reglas*)

acción ← REGLA-ACCIÓN[*regla*]

estado ← ACTUALIZAR-ESTADO(*estado*, *acción*)

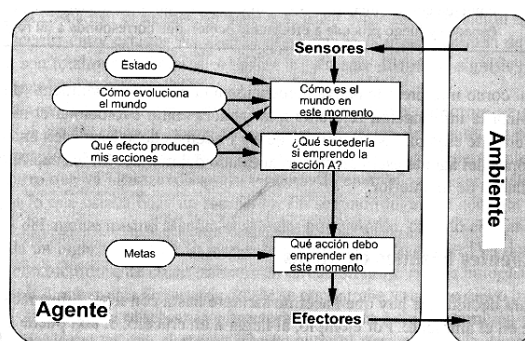
responder con una acción



Tipos IV

Agentes basados en metas

- ✓ Información
 - ✓ Estado del sistema
 - ✓ Situación objetivo
- ✓ Características
 - ✓ Mayor complejidad
 - ✓ Mayor flexibilidad
- ✓ Aplicaciones
 - ✓ Búsqueda en juegos
 - ✓ Planificación

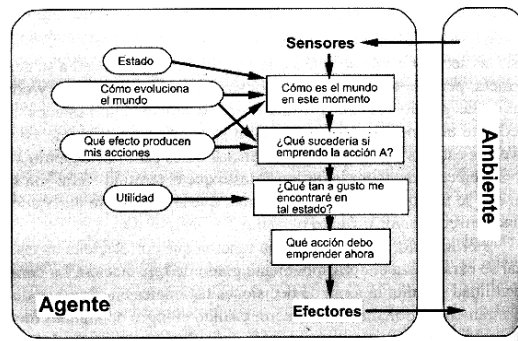




Tipos (y V)

Agentes basados en utilidad

- ✓ Incorporan la utilidad de cada actuación realizada
- ✓ Grado de satisfacción del agente



Estructura I

La estructura de un agente depende del ambiente en el que desempeñe su tarea.

Caracterización

- ✓ Accesibilidad (Detección de todos los aspectos relevantes del estado)
 - ✓ No es necesario mantener una representación de un estado interno
- ✓ Determinista (Estado actual definido por el anterior y las acciones realizadas por el agente)
 - ✓ Manejo de incertidumbre
- ✓ Episódico (Episodio = Agente y actuación. Si la calidad de la actuación depende del episodio y no de los anteriores)
 - ✓ No ha de pensar por adelantado
- ✓ Estáticos (Modificaciones del ambiente durante las deliberaciones)
- ✓ Discreto o Continuo (Respecto a la evolución temporal)



Estructura (y II)

Ejemplos de ambiente

Ambiente	Accesible	Determinista	Episódico	Estático	Discreto
Ajedrez con reloj	Si	Si	No	Semi	Si
Ajedrez sin reloj	Si	Si	No	Si	Si
Póquer	No	No	No	Si	Si
Backgammon	Si	No	No	Si	Si
Conducir un taxi	No	No	No	No	No
Sistema de diag. médico	No	No	No	No	No
Sistema de análisis de imág.	Si	Si	Si	Semi	No
Robot clasificador de partes	No	No	Si	No	No
Controlador de refinería	No	No	No	No	No
Asesor de inglés interactivo	No	No	No	No	Si



Estructura. Ejemplos

Agentes para la solución de problemas

- ✓ Disponen de una meta
 - ✓ Procedimiento de formulación de meta
- ✓ Definición formal de problemas
- ✓ Incorporan procedimientos de búsqueda

Agentes basados en el conocimiento

- ✓ Problemas de definición de KB (Base de Conocimiento)
 - ✓ Se sigue un enfoque declarativo
- ✓ Implementación de mecanismos de inferencia
- ✓ La total independencia requiere de procedimientos de aprendizaje



Ejemplos. Problem Solving I

FORMULAR-META. Ejemplo

✓ Viajero en Cuba que se tiene que marchar a España.

- ✓ Origen: Camagüey
- ✓ Vuelo: Habana-Madrid
- ✓ Salida: Día siguiente
- ✓ Dispone de un coche
- ✓ OBTENIDOS DE LA ACTUALIZACIÓN

✓ Meta:

- ✓ Encontrar una ruta Camagüey-Habana

```
function SIMPLE-PROBLEM-SOLVING-AGENT (p) return acción
input: p, una percepción
static: s, secuencia de acciones, originalmente vacía
        estado, descripción actual del mundo
        g, una meta, originalmente nula
        problema, formulación de un problema

estado ← ACTUALIZAR-ESTADO(estado, p)
if s está vacía then
    g ← FORMULAR-META(estado)
    problema ← FORMULAR-EL-PROBLEMA(estado, g)
    s ← BUSCAR(problema)
acción ← RECOMENDACIÓN(s, estado)
s ← RESTO(s, estado)
return acción
```



Ejemplos. Problem Solving (y II)

Formulación de problemas (FORMULAR-PROBLEMA)

- ✓ Espacio de estados (*estado*)
- ✓ Definición de estado inicial y finales (META *g*)
- ✓ Operaciones
- ✓ Coste de la solución (Heurística)

Sistema de producción (BUSCAR)

- ✓ **Base de datos** (Elementos del espacio de estados), **Reglas de producción** (Operaciones), **Sistema de control** (estrategia de búsqueda)

RECOMENDACIÓN: Empezar la primera acción

RESTO: Elimina la acción de la secuencia



Ejemplos. SBC I

Componentes medulares

- ✓ Base de conocimientos
- ✓ Sistema de inferencia

Necesario

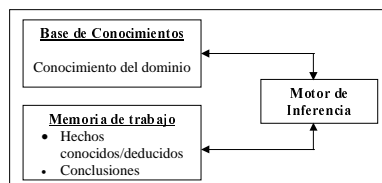
- ✓ Mecanismos de representación del conocimiento
 - ✓ Reglas
 - ✓ Redes Semánticas

```
function KB-AGENT(p) returns acción
input: p, una percepción
static: KB, Base de conocimientos
        t, reloj inicialmente a 0 para medir el tiempo

INFORMAR(KB, HACER-PERCEPCION-ORACION(p, t))
acción ← PREGUNTAR(KB, HACER-CONSULTA(t))
INFORMAR(KB, HACER-ACCION-ORACION(acción, t))
t ← t+1

return acción
```

SISTEMA EXPERTO



Ejemplos. SBC (y II)

Modificaciones de la KB

- ✓ INFORMAR
 - ✓ Incorporación de información (Fórmulas FNC, red semántica)
 - Operación assert (HACER-XXXXX-ORACION)
 - ✓ Procesamiento de la información de entrada
 - HACER-PERCEPCION-ORACION
- ✓ PREGUNTAR
 - ✓ Explotación de la KB (Consultas o "query"s)
 - HACER-CONSULTA
 - ✓ Incorpora el Motor de Inferencia
 - Encadenamiento
 - Manejo de incertidumbre



Ejemplos: El mundo de WUMPUS I

Objetivo

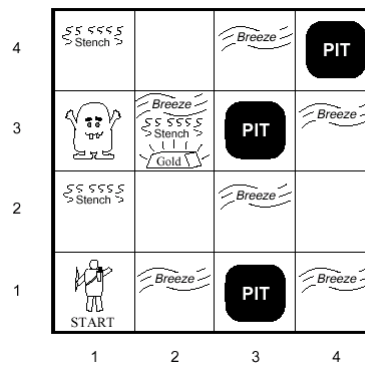
- ✓ Conseguir el oro y llevarlo a casa

Percepciones

- ✓ Brisa (Breeze), Hedor (Stench)
- ✓ Choques

Acciones:

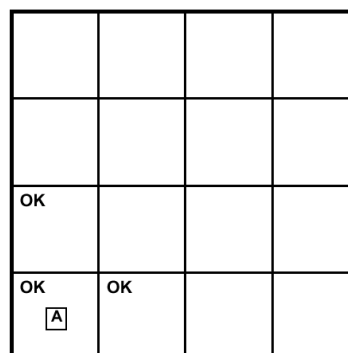
- ✓ Tomar el oro
- ✓ Matar al WUMPUS
- ✓ Saltar (Salir de la cueva)
- ✓ Muerte: Wumpus, Precipicio (PIT)



Ejemplos: El mundo de WUMPUS II

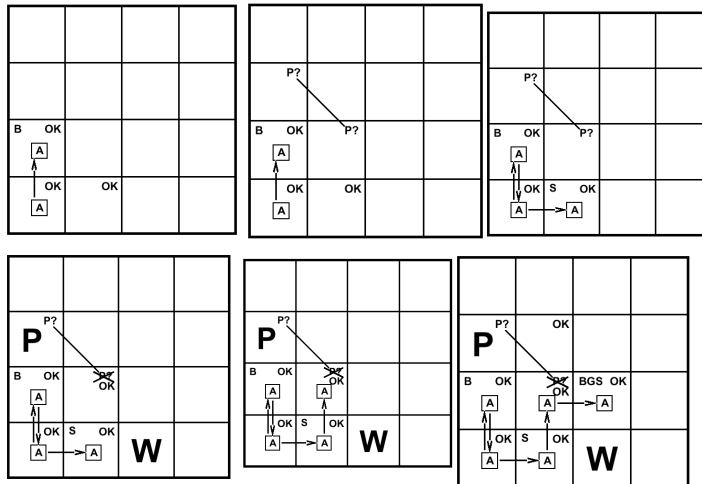
Descripción del ambiente

- ✓ Determinista
- ✓ Accesible
 - ✓ NO, solo tiene una percepción local
 - ✓ Necesita estado interno
- ✓ Estático
 - ✓ El enemigo no se mueve
- ✓ Discreto
 - ✓ No necesita discretizar





Ejemplos: El mundo de WUMPUS III



Evolución

Intel. Artif e Ing. del Conocimiento

Agentes Inteligentes

19

©Vidal Moreno Rodilla. Dpto Inf. y Autom. USAL



Ejemplos: El mundo de WUMPUS IV

Acciones

- ✓ De movimiento: Giro(dcha/izqda), Avanzar, Saltar (Entr/Sal)
- ✓ De adquisición: Coger, Soltar
- ✓ Defensa: Disparar (una vez)

Percepción (fórmula atómica)

- ✓ Percepción([Hedor,Brisa,Resplandor,Golpe,Gemido],t)

Objetivo de Preguntar (explotación de la información)

$$\exists aAcción(a,t)$$

- Se va incorporando información a la base de conocimientos

Intel. Artif e Ing. del Conocimiento

Agentes Inteligentes

20

©Vidal Moreno Rodilla. Dpto Inf. y Autom. USAL



Ejemplos: El mundo de WUMPUS V

Objetivos

- ✓ Generar una base de conocimientos
 - ✓ Basada en lógica de predicados.
 - ✓ Capacidad de representación
- ✓ Motor de inferencia

1ª aproximación. Agente reflejo simple

- ✓ Reglas que conectan percepciones y acciones
 - ✓ Ej: "Si hay resplandor se coge el oro"
 $(\forall h, b, c, g, t) \text{Percepción}(h, b, \text{Resplandor}, c, g, t) \Rightarrow \text{Acción}(\text{Tomar}, t)$
- ✓ Abstracción de la entrada (separación de procedimientos)
 $(\forall b, r, c, g) \text{Percepción}(\text{Hedor}, b, r, c, g) \Rightarrow \text{Hedor}(t)$
 $(\forall t) \text{EnOro}(t) \Rightarrow \text{Acción}(\text{Tomar}, t)$



Ejemplos: El mundo de WUMPUS VI

1ª Aproximación. Agente reflejo

- ✓ Acción Saltar
 - ✓ El agente debe determinar si existe posibilidad de tomar el oro o no
 - ✓ No dispone de percepción sobre el oro, wumpus, precipicios
- ✓ No evitan bucles infinitos

1ª Mejora. Generación de un modelo interno

- ✓ Implementación de todas las percepciones obtenidas
 - ✓ 1ª posibilidad. Histórico de percepciones
 - ➔ 2ª posibilidad. Las percepciones generan un modelo interno de la realidad
- ✓ Reglas diacrónicas
 - ✓ Reflejan la modificación del modelo a lo largo del tiempo
 - Con pérdida de información $En(\text{Agente}, [1,1])$



Ejemplos: El mundo de WUMPUS VII

2ª Mejora. Cálculo de situaciones

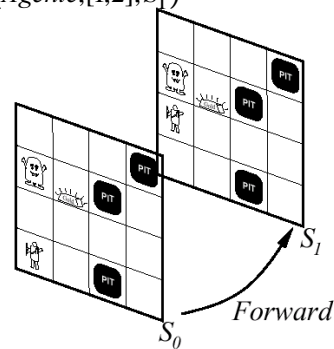
- ✓ Situación: Estructura de datos que refleja la "FOTO" del estado del mundo

$$En(Agente, [1,1], S_0) \wedge En(Agente, [1,2], S_1)$$

- ✓ No todos los predicados necesitan este argumento (Paredes)
- ✓ Aparece un predicado de evolución

$$Resultado(Avanzar^*, S_0) = S_1$$

$$Resultado(Girar(Dcha), S_1) = S_2$$



Ejemplos: El mundo de WUMPUS VIII

2ª Mejora. Cálculo de situaciones

- ✓ Descripción de las acciones
- ✓ Efectos de las acciones (AXIOMAS DE EFECTO)

- Acción Tomar

$$Transportable(Oro)$$

$$(\forall s) EnOro(s) \Rightarrow Pr esente(Oro, s)$$

$$(\forall x, s) [Pr esente(x, s) \wedge Transportable(x)] \Rightarrow Sostener(x, Resultado(Tomar, s))$$

- Acción Soltar

$$(\forall x, s) \neg Sostener(x, Resultado(Soltar, s))$$

- ✓ Efectos que no tienen (AXIOMAS DE CUADRO)

$$(\forall a, x, s) [Sostener(x, s) \wedge a \neq Soltar] \Rightarrow Sostener(x, Resultado(a, s))$$

$$(\forall a, x, s) [\neg Sostener(x, s) \wedge a \neq Tomar \vee [\neg Pr esente(x, s) \wedge Transportable(x)]] \Rightarrow \neg Sostener(x, Resultado(a, s))$$

- ✓ Problema

- Casuística compleja (oro resbaladizo, manos con polvo, etc)



Ejemplos: El mundo de WUMPUS IX

2ª Mejora. Cálculo de situaciones

- ✓ Descripción de las acciones. AXIOMAS DE ESTADO SUCESOR

$$[Verdadera\ después] \Leftrightarrow [Acción\ la\ válida] \vee [Sigue\ siendo\ válida \wedge Ninguna\ acción\ la\ hace\ falsa]$$

$$(\forall a, x, s) Sostener(x, Resultado(a, s)) \Leftrightarrow [a = Tomar \wedge Presente(x) \wedge Transportable(s)] \vee [Sostener(x, s) \wedge a \neq Soltar]$$

- ✓ Falta por determinar los efectos de las acciones en la ubicación
- ✓ Conocimiento sobre el dominio (Cercanía de los peligros)

$$\begin{aligned} Orientación(Agente, S_0) &= 0 \\ (\forall x, y) Posiciónhacia([x, y], 0) &= [x + 1, y] \\ (\forall p, l, s) En(p, l, s) \Rightarrow Posiciónfrente(p, s) &= Posiciónhacia(l, Orientación(p, s)) \\ (\forall l_1, l_2) Adyacente(l_1, l_2) \Leftrightarrow (\exists d) l_1 &= Posiciónhacia(l_2, d) \end{aligned}$$



Ejemplos: El mundo de WUMPUS X

2ª Mejora. Cálculo de situaciones

- ✓ Definición de la ubicación.
- ✓ Definición de la pared (No depende de la situación ni del tiempo)

$$(\forall x, y) Muro([x, y]) \Leftrightarrow [(x = 0) \vee (x = 5) \vee (y = 0) \vee (y = 5)]$$

- ✓ Efecto sobre las acciones y situaciones

- Acción de Avanzar

$$(\forall a, l, p, s) \quad En(p, l, Resultado(a, s)) \Leftrightarrow \left[\begin{aligned} &[a = Avanzar \wedge l = PosiciónFrente(p, s) \wedge \neg Muro(l)] \vee \\ &\vee (En(p, l, s) \wedge a \neq Avanzar) \end{aligned} \right]$$

- Acción de Girar

$$(\forall a, d, p, s) \quad (Orientación(p, Resultado(a, s)) = d) \Leftrightarrow \left[\begin{aligned} &[a = Giro(Dcha) \wedge d = Mod(Orientación(p, s) - 90, 360)] \vee \\ &\vee [a = Giro(Izqda) \wedge d = Mod(Orientación(p, s) + 90, 360)] \vee \\ &\vee [Orientación(p, s) \wedge \neg [a = Giro(Dcha) \wedge a = Giro(Izqda)]] \end{aligned} \right]$$



Ejemplos: El mundo de WUMPUS XI

- 2ª Mejora. Cálculo de situaciones
 - ✓ Definición de las reglas del Wumpus y la acción de disparar (Ejercicio)
- 3º. Definición de un modelo interno
 - ✓ Implementación de la capacidad de abstracción de la información
 - ✓ Modelo = Posición + Conocimiento cualitativo. (Propiedades ocultas)
 - ✓ Percepción --> Asociación a un lugar de la percepción

$$(\forall l, s) [En(Agente, l, s) \wedge Brisa(s)] \Rightarrow Airosol(l)$$

$$(\forall l, s) [En(Agente, l, s) \wedge Hedor(s)] \Rightarrow Oloroso(l)$$
 - ✓ Reglas Sincrónicas: Reflejan deducciones en un instante de tiempo que permiten generar un modelo interno perfeccionado



Ejemplos: El mundo de WUMPUS XII

- 3º. Definición de un modelo interno
 - ✓ Reglas sincrónicas (Captura de información en un instante de tiempo)
 - ✓ Reglas causales
 - Propiedades ocultas provocan percepciones
 - Sistemas de razonamiento basados en este tipo de reglas
 - **Sistemas basados en modelos**
 - Ejemplo

$$(\forall l_1, l_2, s) [En(Wumpus, l_1, s) \wedge Adyacente(l_1, l_2)] \Rightarrow Oloroso(l_2)$$

$$(\forall l_1, l_2, s) [Pr ecipicio(l_1) \wedge Adyacente(l_1, l_2)] \Rightarrow Airosol(l_2)$$
 - Regla basada en modelo

$$(\forall l, t, s) [\neg En(Wumpus, l, s) \wedge \neg Pr ecipio(l)] \Leftrightarrow OK(l)$$



Ejemplos: El mundo de WUMPUS XIII

3º. Definición de un modelo interno. Reglas sincrónicas

✓ Reglas de diagnóstico

- Percepciones permiten inferir propiedades ocultas

– Considerar $(\forall l, s)[En(Agente, l, s) \wedge Brisa(s)] \Rightarrow Airoso(l)$
 $(\forall l, s)[En(Agente, l, s) \wedge Hedor(s)] \Rightarrow Oloroso(l)$

– En el ejemplo

$(\forall l, s) Oloroso(l_1) \Rightarrow [(\exists l_2) En(Wumpus, l_2, s) \wedge ((l_2 = l_1) \vee Adyacente(l_1, l_2))]$

– Se obtiene información de forma directa
 – Conclusiones pobres

- Comparación del modelo diagnóstico

$(\forall l_1, l_2, g, u, c) Percepción([Nada, Nada, g, u, c], t) \wedge$
 $\wedge En(Agente, l_1, s) \wedge Adyacente(l_1, l_2) \Rightarrow OK(l_2)$

– Frente a la regla causal

$(\forall l, t, s) [\neg En(Wumpus, l, s) \wedge Precipio(l)] \Leftrightarrow OK(l)$



Ejemplos: El mundo de WUMPUS XIV

Sistema basado en modelo (reglas causales) vs. Sistemas diagnóstico

✓ Considerar

$(\forall l_1, l_2, g, u, c) Percepción([Nada, Nada, g, u, c], t) \wedge$
 $\wedge En(Agente, l_1, s) \wedge Adyacente(l_1, l_2) \Rightarrow OK(l_2)$

$(\forall l, t, s) [\neg En(Wumpus, l, s) \wedge Precipio(l)] \Leftrightarrow OK(l)$

Existen cuadros seguros con Olor o Brisa

En áreas de aplicación de IA

✓ Medicina

- ✓ Método diagnóstico (Asociación directa Síntoma <-> Enfermedad) esta siendo reemplazado por

- Utilización de un modelo explícito de la enfermedad y cómo se manifiestan sus síntomas
 - Enfermedad -> Verificación de síntomas



Ejemplos: El mundo de WUMPUS XV

✎ Incorporación de la utilidad

- ✓ Se implementan grados de satisfacción en la selección de una acción
- ✓ Se separan los hechos que hacen referencia a los objetivos (Metas) de los que afectan a las Acciones (Modularidad en la valoración)
- ✓ Tipos de acción: Estupenda, Buena, Regular, Mala

$$(\forall a,s)Estupenda(a,s) \Rightarrow Acción(a)$$

$$(\forall a,s)[Buena(a,s) \wedge (\neg \exists b)Estupenda(b,s)] \Rightarrow Acción(a)$$

$$(\forall a,s)[Regular(a,s) \wedge (\neg \exists b)[Estupenda(b,s) \vee Buena(b,s)]] \Rightarrow Acción(a)$$

$$(\forall a,s)[Mala(a,s) \wedge (\neg \exists b)[Estupenda(b,s) \vee Buena(b,s) \vee OK(b,s)]] \Rightarrow Acción(a)$$



Ejemplos: El mundo de WUMPUS XVI

✎ Incorporación de la utilidad

- ✓ Mundo de WUMPUS
 - ✓ Excelente
 - Tomar el Oro y salir
 - ✓ Buena
 - Estar en un estado OK y todavía no visitado
 - ✓ Regular
 - Estar en un estado OK y YA visitado
 - ✓ Mala
 - Cuadro No Mortal pero sin situación de OK
 - ✓ Fatal
 - Muerte súbita

✎ FALTA POR DEFINIR UNA META AL SISTEMA



Ejemplos: El mundo de WUMPUS XVII

- ☞ La información anterior permite alcanzar el objetivo
 - ✓ La utilidad representa a la información local (Heurística)
 - ✓ Las metas representan políticas de funcionamiento
$$(\forall s) \text{Sostener}(\text{Oro}, s) \Rightarrow \text{PosiciónMeta}([1,1], s)$$
- ☞ Una meta explícita permite/obliga diseñar la secuencia de acciones que permiten alcanzarla
 - ✓ Procedimientos
 - ✓ Inferencia. En sistemas basados en reglas a veces es posible PREGUNTAR la secuencia de acciones (Con los estados visitados)
 - ✓ Búsqueda. Utilización de un procedimiento de búsqueda (Mejor-Primero). Se traduce el conocimiento en operadores y la situación en un estado
 - ✓ Planificación: Sistemas de razonamiento de propósito especial que permiten razonar acerca de las acciones



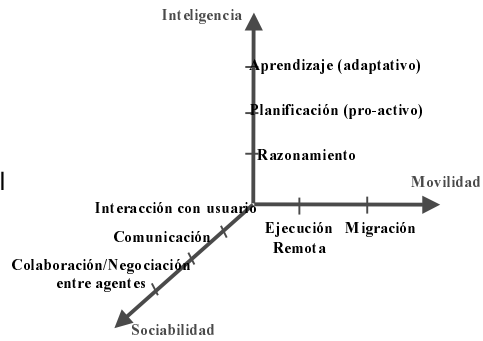
Ejemplos: El mundo de WUMPUS XVIII

- ☞ Sistemas de planificación
 - ✓ Plan: Secuencia de acciones
 - ✓ Es distinto a una estrategia de búsqueda
 - Exploración exhaustiva (Ej: Grafos)
 - Información local
 - ✓ En el ejemplo
 - ✓ Dada la pregunta
$$\text{PREGUNTAR}(\text{KB}, (\exists p) \text{Sostener}(\text{Oro}, \text{PlanResult}(p, S_0)))$$
 - ✓ se obtiene (Oro en [1,2]) {p., [Adelante, Coger]}
 - ✓ Definición
$$(\forall s) \text{PlanResult}([], s) = s$$
$$(\forall a, p, s) \text{PlanResult}([a | p], s) = \text{PlanResult}(p, \text{Resultado}(a, s))$$



Entornos de utilización

- ❏ Ejemplos de entornos
 - ✓ Sistemas de gestión
 - ✓ Internet
 - ✓ Agentes de interfaz
 - ✓ Control Industrial
 - ✓ Agentes en tiempo real
- ❏ Características
 - ✓ Necesidad de comunicación y negociación
 - ✓ Movilidad



Agentes Móviles⁽¹⁾

- ❏ Concepto
 - ✓ Taxonomía del código móvil
 - ✓ AM vs RPC
- ❏ Ventajas
- ❏ Aplicaciones
 - ✓ Implementación
 - ✓ Sistemas comerciales

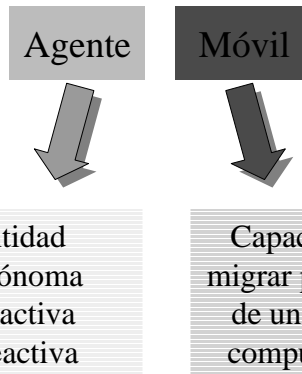
(1). Agradecimiento a J. A Pavón. UPM "Agentes Móviles en Internet"



Conceptos

➤ Agregación de objetivos

- ✓ Inteligencia
- ✓ Autonomía
- ✓ Movilidad



Conceptos

- Agentes móviles y Agentes inteligentes
- Los agentes inteligentes son capaces de ofrecer un comportamiento "inteligente": razonamiento, planificación, aprendizaje, etc.
- Los agentes móviles (AM) tienen la capacidad de moverse por distintos nodos de una red (una o más veces)
 - ✓ perspectiva de sistema distribuido
 - ✓ aspectos táctiles: movilidad, persistencia
 - ✓ y (en parte) de sociabilidad: comunicación y colaboración
 - ✓ NO se consideran fundamentales los aspectos cognitivos (adaptación, aprendizaje y planificación)
 - ✓ Para la comunidad de IA la movilidad no es un atributo de los agentes
 - ✓ Para la comunidad de AM la movilidad es el atributo principal



Conceptos

Agente móvil

- ✓ Realiza sus tareas no sólo en la computadora de su propietario, sino también en otras en la red
 - ✓ buscando información en beneficio de su propietario o negociando y cerrando tratos en su nombre o utilizando servicios remotos
- ✓ Tiene capacidad para decidir a qué servidores moverse
 - ✓ Hay instrucciones explícitas para que el agente pueda parar su ejecución, migrar a otro nodo (preservando su estado), y continuar su ejecución
- ✓ Puede moverse a uno o más servidores
- ✓ Es una extensión del modelo cliente-servidor
 - ✓ Los clientes envían parte de ellos al servidor (o a varios servidores) para ejecutarse



Taxonomía del código móvil

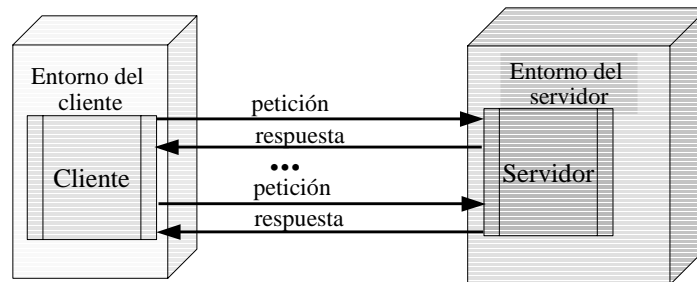
Dependiente de la aplicación

- ✓ *applet* aplicación cargada por la red para ejecutar localmente
- ✓ *servlet* agente que es cargado en un lugar remoto donde será activado como un servidor
- ✓ *piglet* agente malicioso que supone riesgo de seguridad



AM vs. RPC

- ❏ Llamada a procedimiento remoto (RPC)
 - ✓ La red pasa cada petición por separado, ... y devuelve una respuesta por separado
 - ✓ Hay acuerdo en los procedimientos, sus parámetros y tipo de resultados
 - ✓ Puede ocasionar mucho tráfico de red



Intel. Artif e Ing. del Conocimiento

Agentes Inteligentes

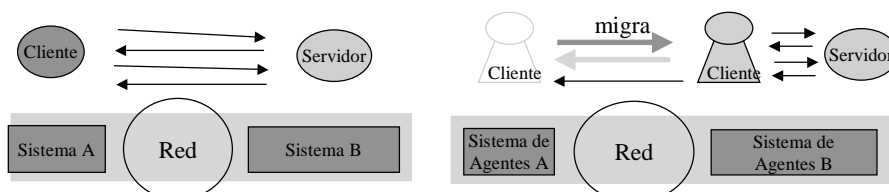
41

©Vidal Moreno Rodilla. Dpto Inf. y Autom. USAL



Agentes móviles: ventajas

- ❏ Reducción del tráfico en la red y de la capacidad de cómputo del cliente
 - ✓ Interesante para entornos de bajo ancho de banda y clientes sencillos
 - ✓ Realiza las interacciones localmente en el sistema objetivo (servidores potentes)
 - ✓ Recuperación y filtrado de la información en su origen
 - ✓ Sólo se devuelven los resultados definitivos (mediante migración del agente o usando mensajes/RPC)



Intel. Artif e Ing. del Conocimiento

Agentes Inteligentes

42

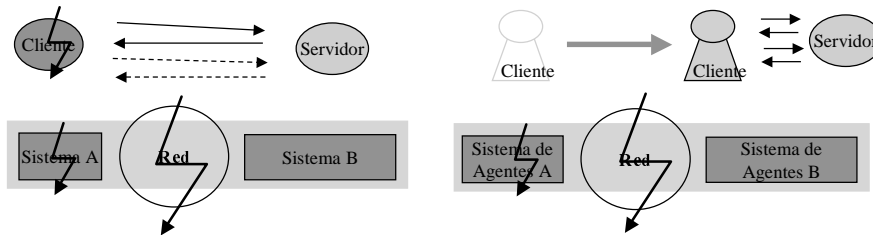
©Vidal Moreno Rodilla. Dpto Inf. y Autom. USAL



Agentes móviles: ventajas

Robusto: reducción de la dependencia de la disponibilidad de la red y del cliente/servidor

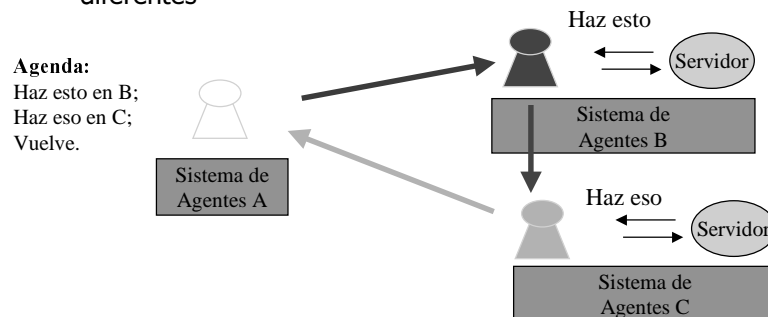
- ✓ los Agentes Móviles migrados al sistema servidor no se ven afectados por los fallos del cliente o de la red



Agentes móviles: ventajas

Automatización del proceso de tareas distribuidas

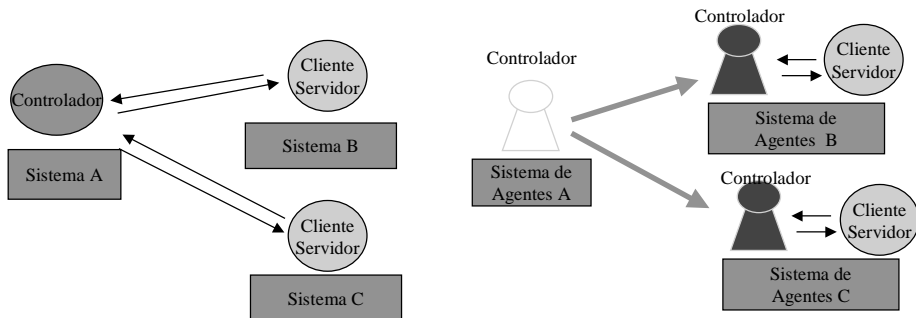
- ✓ Los Agentes Móviles realizan tareas específicas en lugares diferentes





Agentes móviles: ventajas

- ❏ Proceso de tareas local/descentralizado
 - ✓ mayor eficiencia, fiabilidad, seguridad, etc.



Agentes móviles: cuestiones

- ❏ Seguridad
- ❏ Detección de virus
 - ✓ ¿Se puede confiar en el agente?
- ❏ Autenticación del usuario
 - ✓ ¿Quién envía el agente móvil?
- ❏ Autenticación del servidor o entorno de ejecución de agentes
 - ✓ ¿No caerá el agente en una trampa?
- ❏ Derechos de ejecución de agentes de un usuario en un servidor
 - ✓ ¿Pueden ejecutarse los agentes? ¿Qué funciones pueden realizar?
- ❏ Capacidad del agente para pagar por los servicios utilizados
 - ✓ Teleclicks (General Magic)



Aplicaciones

- ❏ Servicios de información en Internet
 - ✓ Recuperación y extracción de información de múltiples lugares
 - ✓ Búsqueda y filtrado de la información
 - ✓ Control de cambios
 - ✓ Difusión de información
- ❏ Comercio electrónico
 - ✓ Mercado de servicios electrónico
 - ✓ Negociación
- ❏ Equipos móviles y PC's en el hogar
 - ✓ Conexiones intermitentes y bajo ancho de banda
- ❏ Redes públicas de telecomunicaciones
 - ✓ Provisión de servicios bajo demanda
 - ✓ Descentralización del control y gestión de redes
- ❏ Procesamiento paralelo
- ❏ Gestión de procesos (workflow)
- ❏ Juegos (agentes que representan jugadores)

Personalización de servicios

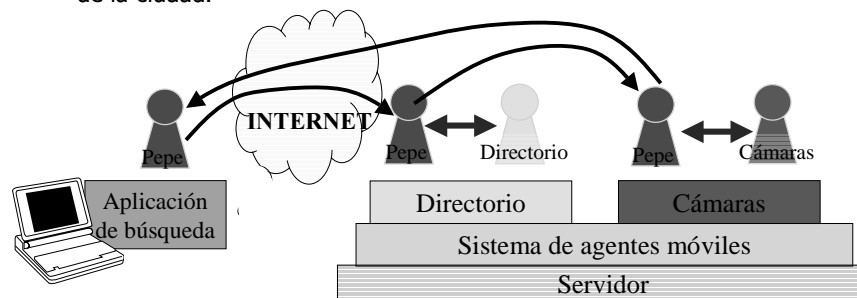
Flexibilidad de la distribución

Delegación de tareas



Aplicaciones: Internet

- ❏ Ejemplo (General Magic): Buscando la cámara de fotos más barata
 - ✓ Tras decidirse por el modelo X, Pepe le solicita a su comunicador personal que le busque dónde comprarla. En 15 minutos tiene los nombres, direcciones y teléfonos de las tres tiendas de su ciudad con los precios más bajos. Puede comprobar que se ahorra 10.000 Ptas. sobre el precio de la tienda más cercana en una tienda de otro barrio de la ciudad.

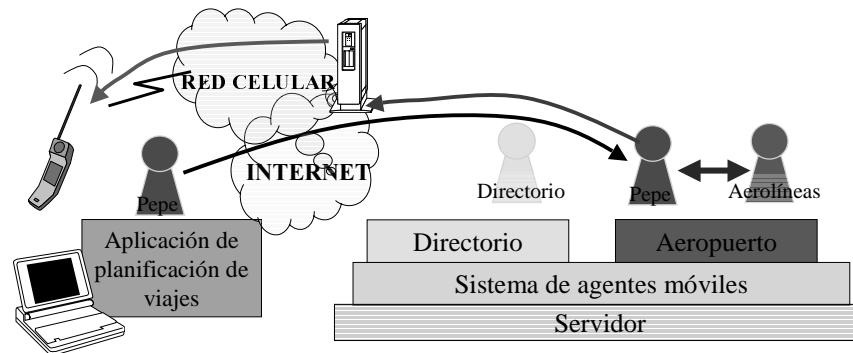




Aplicaciones: Internet

Ejemplo (General Magic): Control de cambios

- ✓ Pepe hizo sus reservas de avión para Santiago hace dos semanas. El día de vuelta se encuentra en una reunión importante de última hora, con el tiempo ajustado. En su móvil recibe un mensaje de que el avión se ha retrasado una hora, y gracias a ello puede dedicar más tiempo a la reunión.



Requisitos plataforma AM

Requisitos del modelo de agentes

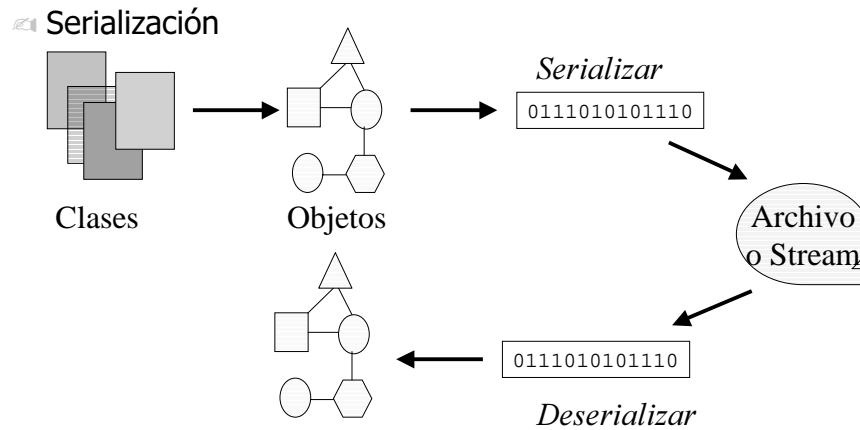
- ✓ Todo agente software está definido por
 - ✓ Modelos de: ciclo de vida, computacional, de seguridad, de comunicación
- ✓ Los agentes móviles añaden
 - ✓ Modelo de *navegación*

Requisitos genéricos

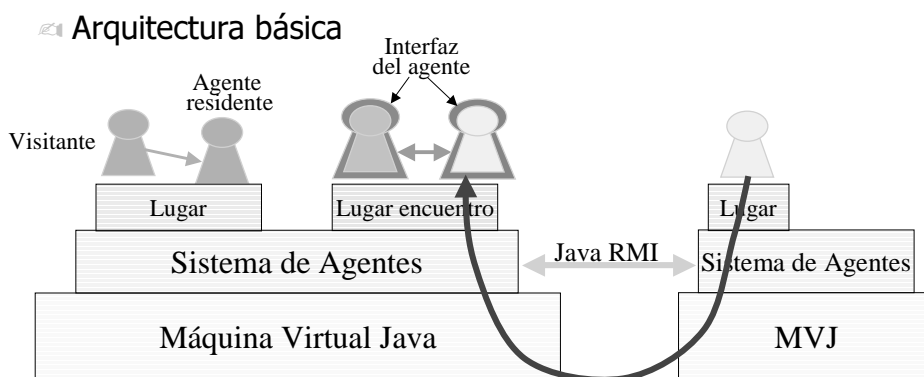
- ✓ Rendimiento y eficiencia
- ✓ Portabilidad
- ✓ Abierto
- ✓ Integración y soporte de sistemas propietarios
- ✓ Escalabilidad
- ✓ Mantenibilidad



Implementación de AM con Java



Implementación de AM con Java





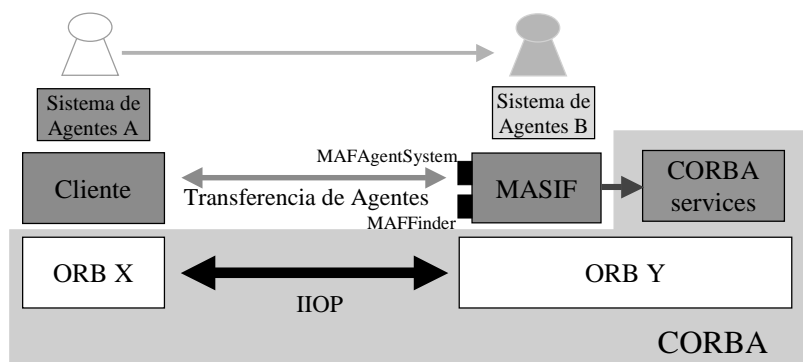
Sistemas de AM comerciales

- General Magic Telescript y Odyssey
✓ <http://www.genmagic.com/agents>
- IBM Aglets
✓ <http://www.trl.ibm.co.jp/aglets>
- Mitsubishi Concordia
✓ <http://www.meitca.com/HSL/Projects/Concordia>
- ObjectSpace Voyager
✓ <http://www.objectspace.com/Voyager>
- Oracle Mobile Agents
✓ http://www.oracle.com/products/networking/mobile_agents.html



MASIF

AM en la arquitectura CORBA





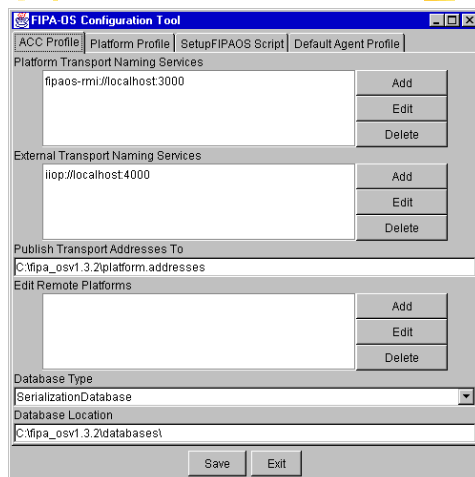
FIPA

- ❏ Foundation for Intelligent Physical Agents (<http://www.fipa.org>)
 - ✓ Consorcio industrial fundado en 1996
 - ✓ Varias decenas de compañías de telecomunicaciones e informática
- ❏ 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 agentes
 - ✓ Selección y adaptación de tecnologías existentes



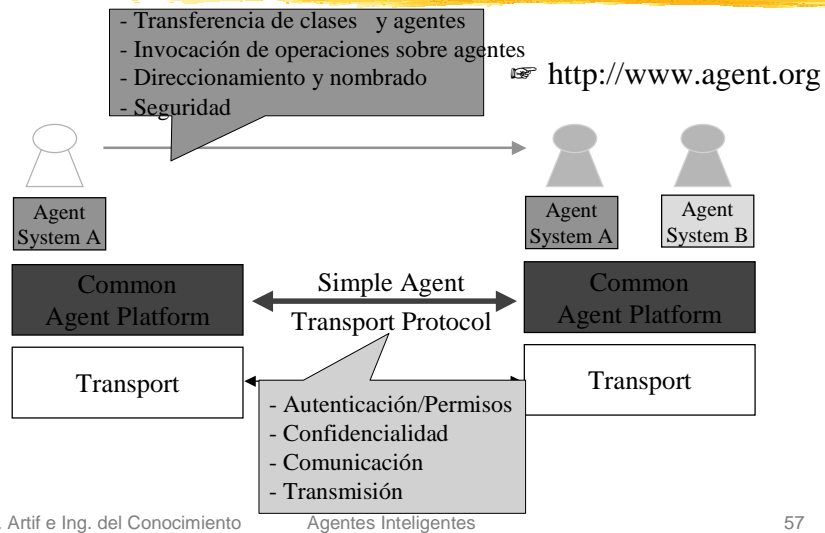
FIPA

- ❏ Instalación
 - ✓ Disponible máquina Java (1.2.2)
 - ✓ Distribución .jar
 - ✓ Dispone de servicios de nombre
- ❏ Soporta diferentes plataformas





Agent Society. Arquitectura de plataforma común de agentes



Conclusiones

- Los AM ofrecen
 - ✓ un marco abierto para la implantación y personalización de servicios en red
 - siguiendo un modelo antropomórfico (fácil de modelar)
 - ✓ capacidad de desarrollar nuevos servicios (luego nuevas posibilidades de negocios en la red)
 - comercio electrónico integral
- Potencian al usuario individual
 - ✓ especialmente en entornos como Internet
 - ✓ el usuario gana en capacidad de cómputo, automatización de tareas distribuidas (por ejemplo, búsqueda y filtrado de información), menor dependencia del acceso a la red



Conclusiones

Estado de la tecnología

- ✓ No hay todavía plataformas de AM fiables
- ✓ ... aunque se está progresando (lentamente)
 - Hay que ver cómo ofrecer un valor añadido aun restringiendo funcionalidad

No hay un uso amplio de plataformas de AM

- Necesidad de definir servicios mezcla de AM/tradicionales
 - Los primeros servicios podrían explotar las capacidades de captura de información avanzadas de los AM
 - Luego pasar a servicios más complejos (negociación, etc.)
- El uso de interfaces estándar como OMG MASIF puede asegurar la interoperabilidad y la apertura en el futuro



Bibliografía

Enlaces

- ✓ http://www.cetus-links.org/oo_mobile_agents.html
- ✓ <http://www.infosys.tuwien.ac.at/Research/Agents/>
- ✓ <http://www.fipa.org>
- ✓ <http://www.javasoft.com>