Introducción al concepto de agente

Sistemas Multi-Agente y Sistemas Autónomos

Juan A. Botía

Departamento de Ingeniería de la Información y las Comunicaciones, Universidad de Murcia

October 3, 2007

Mejor con un ejemplo

- no existe una definición clara y consensuada
- texto está extraido de [6]:

> *"Imagine su propio mayordomo móvil, capaz de viajar con usted y organizar cada uno de los asuntos cotidianos de su vida, desde las reuniones que ha de tener hasta los restaurantes en los que va a comer.[] El programa, basado en inteligencia artificial corre en teléfonos móviles y es capaz de determinar las preferencias de usuario y usar la Web para planificar negocios y eventos sociales.* El primer dia que el mayordomo comienza a trabajar, se comportará de manera educada al no saber demasiado sobre el usuario pero a medida que trabajamos juntos, estará más y más familiarizado con mis preferencias y será capaz de tomar decisiones sin tener que consultarme, *dice el profesor Jennings. Los algoritmos de aprendizaje permitirán al mayordomo el organizar reuniones sin la necesidad de consultar constantemente al usuario [] diseñado para trabajar con un tipo de teléfonos móviles que están saliendo ahora al mercado"*

Asistente personal para crear historias gráficas

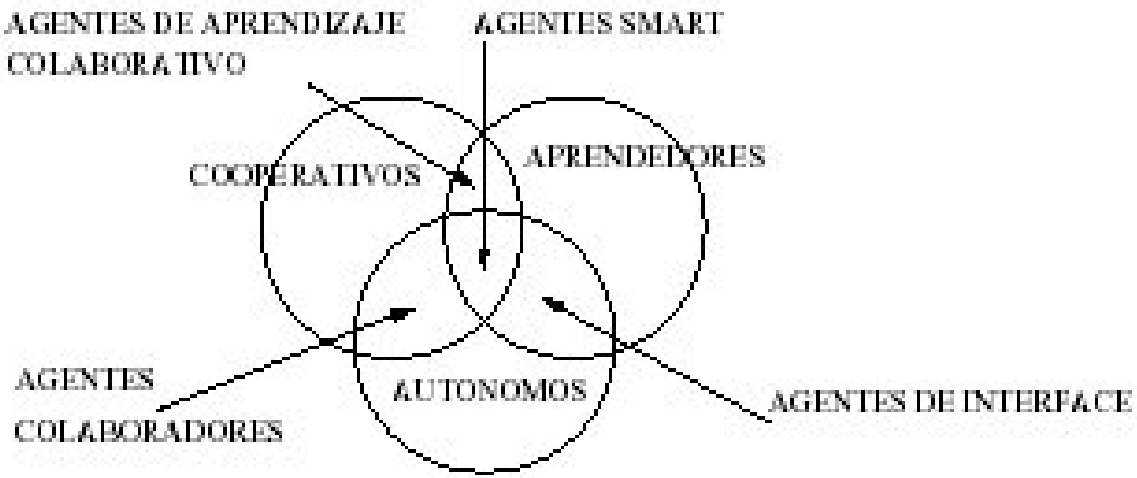
Características que hacen a ARIA diferente

- No es una aplicación pasiva (no hace nada hasta que el usuario se lo ordena) sino que es **proactiva**
- No es una aplicación estática (capacidad de adaptación)
  - adaptación se consigue mediante la deducción, un tipo de **aprendizaje automático**
- La aplicación **colabora con el usuario**

Clasificaciones de agentes

Clasificación de Nwana [8]

- Dependiendo de las dimensiones a través de las cuales intentemos clasificar a los agentes, vamos a tener una determinada tipología.
- Nwana [8] usa dimensiones como la movilidad, si son deliberativos o reactivos o en base a unos cuantos atributos ideales como son la autonomía, cooperación y capacidad de aprendizaje.



Agents are Everywhere

Dos perspectivas diferentes

Usuario final

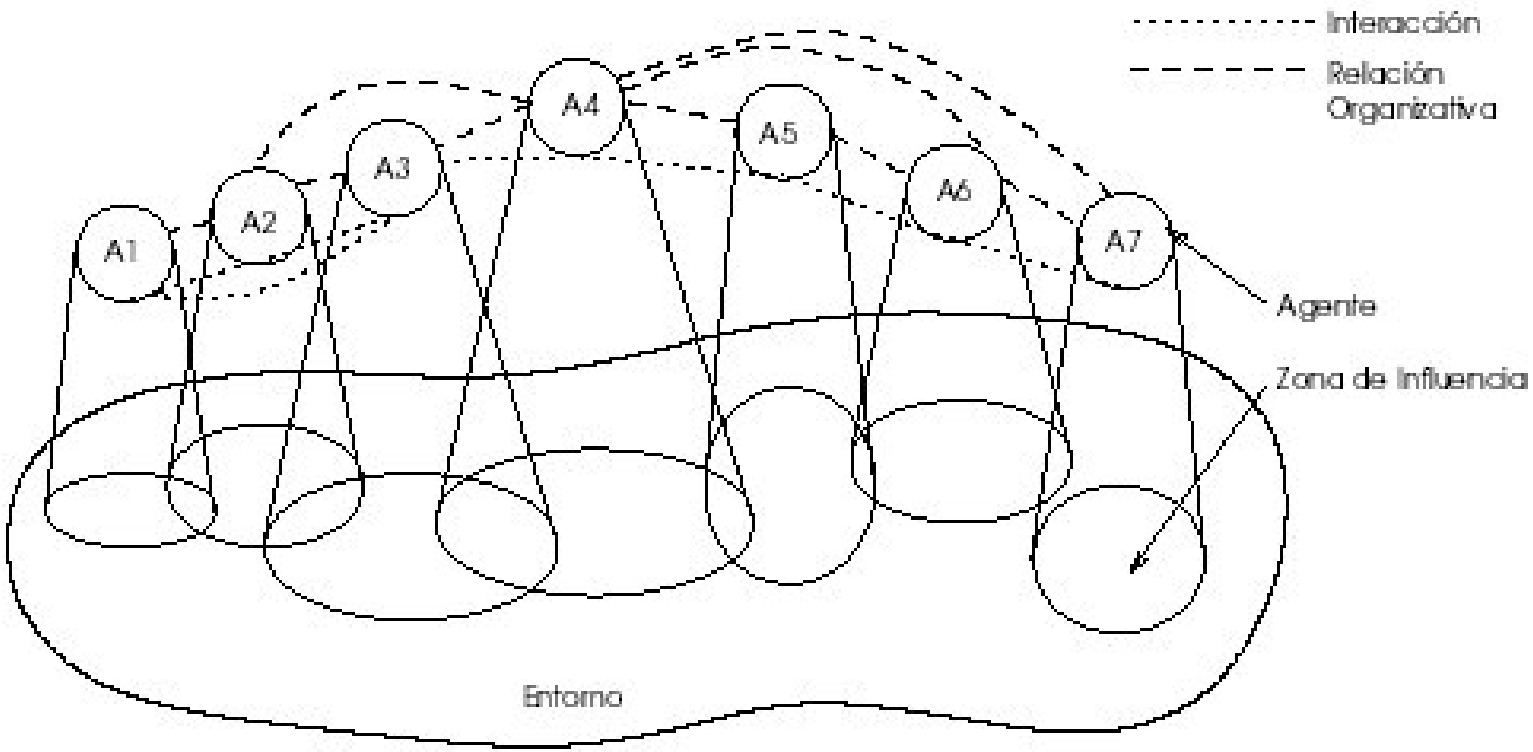
- Gestionan carteras de inversión
- Organizan reuniones de trabajo
- Hacen las compras en Internet por nosotros
- Filtran la información que nos interesa

Ingeniero de software

- Potente metáfora para especificación de sistemas complejos
- Metodologías válidas (análisis y diseño)
- Herramientas para implementación de MAS están disponibles

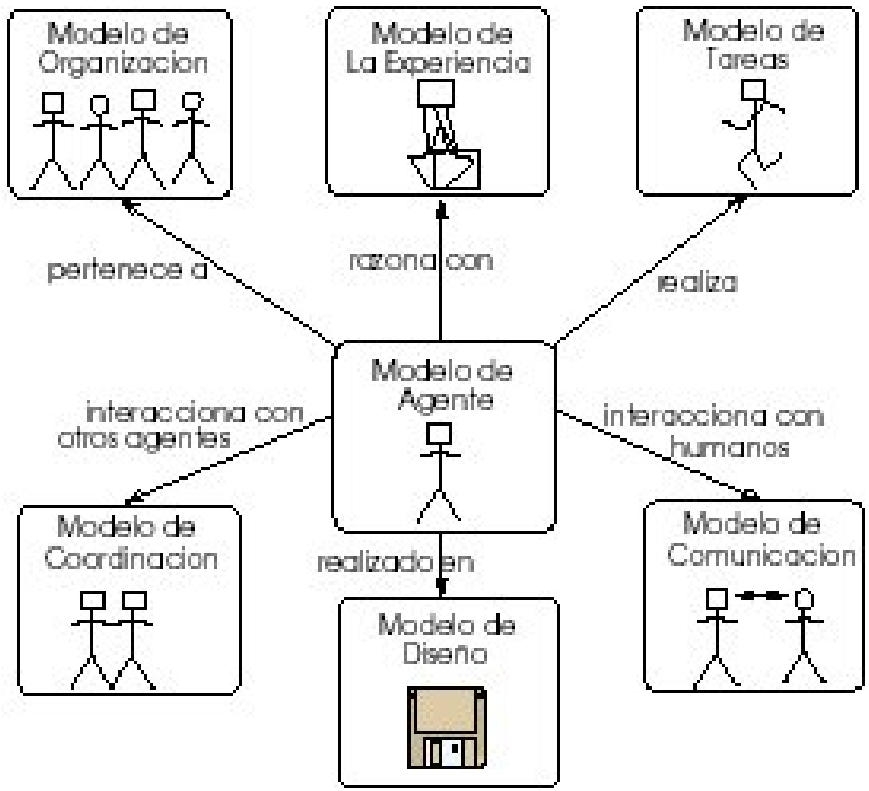
Perspectiva ingenieril

La metáfora de agente y MAS (Jennings y Wooldridge, 2001)



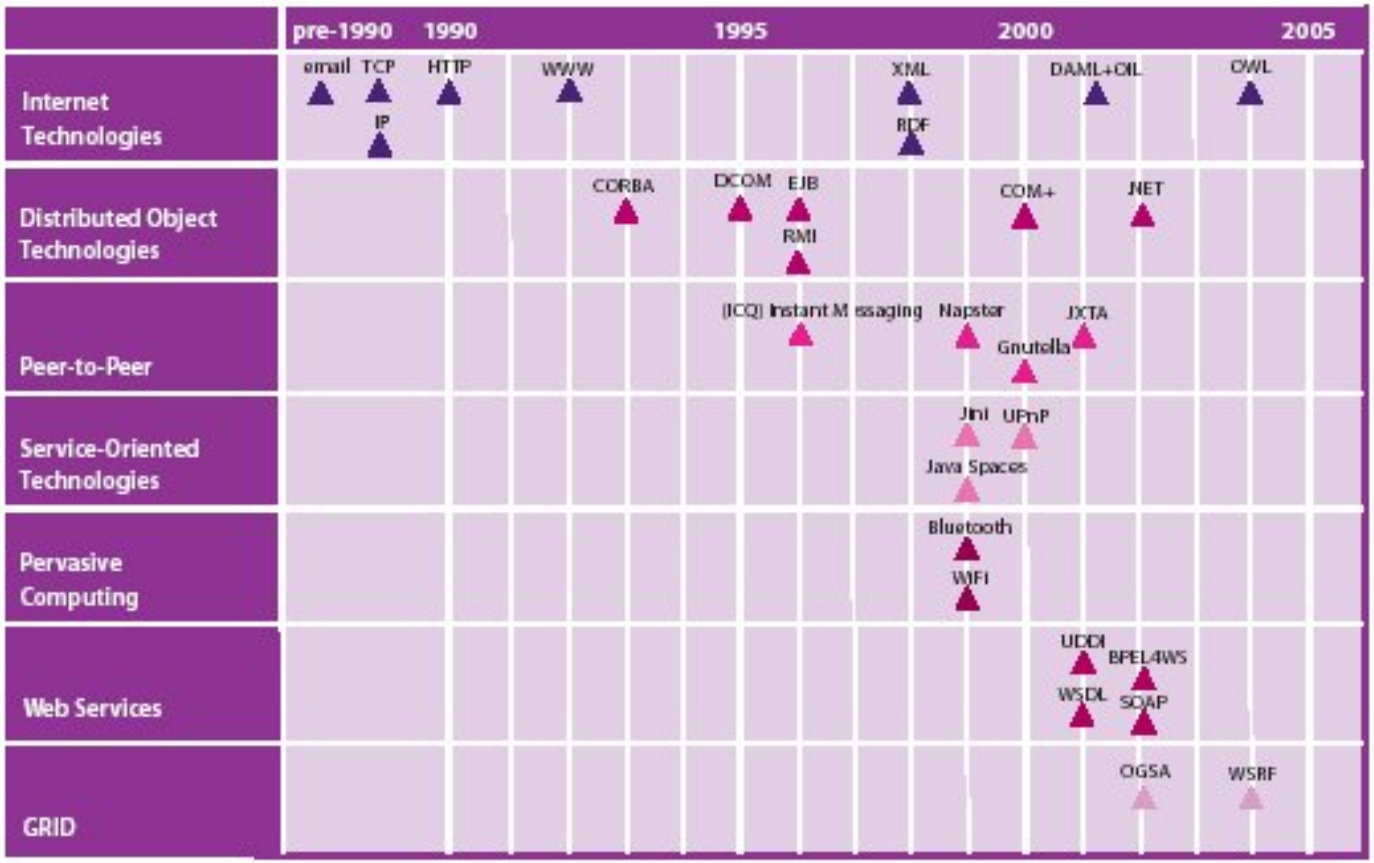
Perspectiva ingenieril

Metodologías de agentes, como MAS-CommonKADS (Iglesias, 1998)



Perspectiva ingenieril

Tecnologías para la construcción



Teorías, arquitecturas y lenguajes de agentes

- Agentes BDI
  - Razonamiento práctico
  - Intenciones en el razonamiento práctico
  - Programación de agentes BDI
- Agentes reactivos
- Agentes híbridos

Elementos en programación de agentes

- Elementos en la programación OO son: clases, objetos, tipos primitivos, métodos, herencia, interfaces, ...
- Elementos en la programación de agentes: Shoham [11]

  "los agentes resultan entidades computacionales que poseen versiones formales de estados mentales y en particular versiones formales de creencias, habilidades, obligaciones y, posiblemente, otras cualidades mentales"

Agent0

El lenguaje diseñado por Shoham se denominó `Agent0`. En éste, un agente consta de

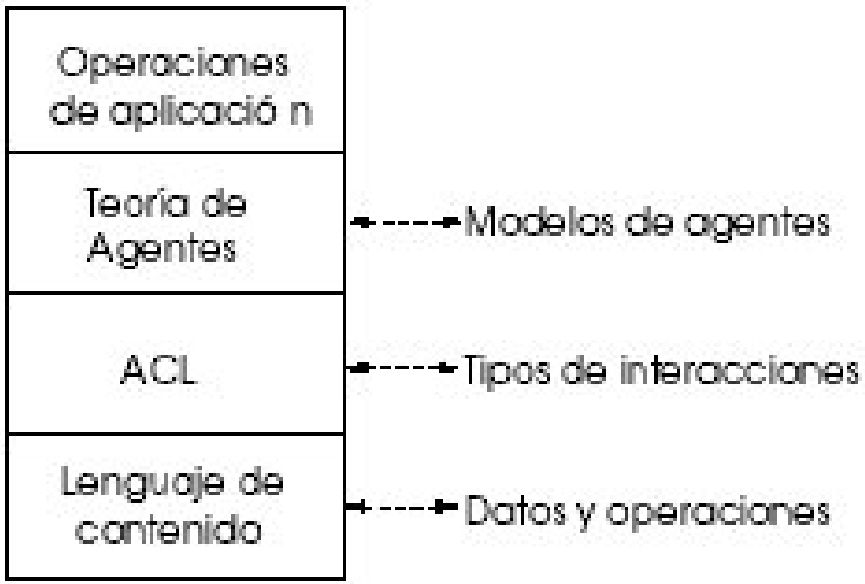
- Creencias: sentencias lógicas sobre el mundo que conoce, que pueden cambiar en el tiempo.
- Obligaciones (i.e. *commitments*): sentencias lógicas como las anteriores, que se refieren a la ejecución de acciones pendientes de ejecución.
- Reglas de obligaciones
  - Antecedente: posibles mensajes a recibir y estados mentales
  - Consecuente acciones a realizar
- Capacidades: acciones que puede llevar a cabo el agente

Ejemplo en Agent0

Así, un ejemplo de regla de obligación codificada en Agent0 puede ser el siguiente:

```
(COMMIT
  ( agent, REQUEST, DO(time, action)),
  (B, [now, Friend agent] AND
      CAN(self, action) AND
      NOT [time, CMT(self, anyaction)]),
  self,
  DO (time, action))
```

Programación por capas



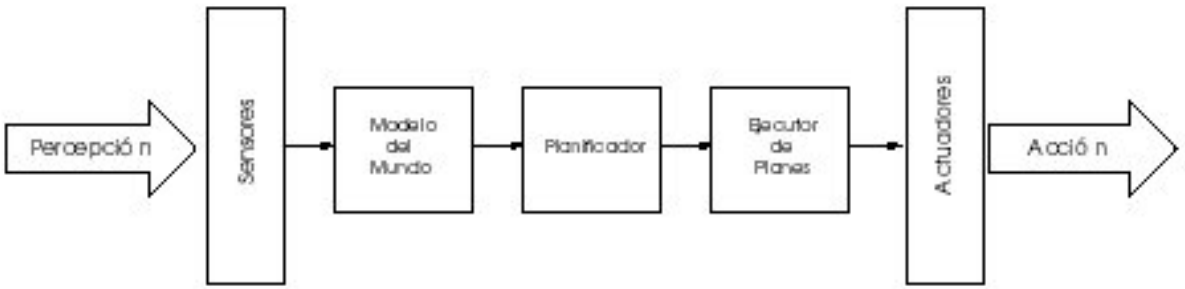
Agentes deliberativos y reactivos

Un agente deliberativo es aquel que "contiene un modelo simbólico del mundo representado internamente de forma explícita y que toma decisiones mediante razonamiento lógico" *[12].*

Un agente reactivo es aquel en que "no incluye un modelo simbólico del mundo ni usa razonamiento simbólico complejo de ningún tipo" *[12].*

Agentes deliberativos

- Funcionan siguiendo el paradigma de los sistemas clásicos planficación de la IA, basado en el ciclo percepción-planificación-acción.



Agentes BDI

- El modelo de agente más representativo del tipo de los deliberativos
- Tienen "ciertas actitudes mentales" [9]
  - **creencias**: conocimiento sobre el resto del mundo $\equiv$ un conjunto de variables, una BD, un conjunto de expresiones lógicas...
  - **deseos**: cómo se ordenan por prioridad los objetivos del agente $\equiv$ estructura de lista ordenada
  - **intenciones**: Cuando el agente actúa sobre el entorno, la acción escogida determina la intención que inmediatamente manifiesta el agente $\equiv$ estructura de datos con la última acción o secuencia de acciones ejecutadas

El razonamiento práctico (i.e. dirigido a las acciones)

> *El razonamiento práctico tiene que ver con la ponderación de consideraciones en conflicto para y contra opciones competidoras, en donde las consideraciones de importancia se obtienen a partir de lo que el agente desea y lo que el agente cree.*

(Bratman, 1990 [1])

Razonamiento teórico: aplicar el modus ponens

Razonamiento práctico: decidir si tomar un tren o un bus para llegar al trabajo

Razonamiento práctico

Partes del RP

1. determinar qué situación ambiental (*estate of affairs*) queremos conseguir (i.e. deliberación)
2. determinar cómo vamos a conseguir la situación ambiental deseada (i.e. razonamiento medios-fines).

Complicaciones en el RP

Problemas en el mundo real

- Recursos limitados (cálculo y memoria)
- Restricciones en el problema
  - Temporales $\longrightarrow$ número de ciclos de CPU finito
- De lo que se deriva un compromiso entre capacidad de reacción y calidad de las decisiones

Intenciones en el razonamiento práctico

- Los humanos manejamos dos tipos de intenciones
  - Las acciones (empujar a alguien bajo un tren con la intención de matarlo)
  - Estados mentales (tener por la mañana la intención de empujar a alguien bajo un tren por la tarde)
- Las que utilizan los agentes son las Las intenciones que caracterizan estados mentales $\longrightarrow$ *intenciones directas futuras*

Intenciones en el razonamiento medios-fines

- Las pro-actitudes (i.e. *pro-attitudes*) son intenciones que pueden dar lugar a acciones.
- E.g. si teneis intención de aprobar el curso, yo puedo esperar de vosotros que tomeis un conjunto *razonable de acciones* para conseguirlo.
- En este sentido, las intenciones juegan un papel importante en la producción de acciones.

Las intenciones persisten

- Un agente se compromete con las intenciones adoptadas
- Si inmediatamente después de comprometerse la rechaza antes de dedicarle algún recurso su comportamiento **no será racional**
- Incluso después de un fallo, volverá a intentarlo
- Podrá abandonarla cuando
  1. Cuando tiene evidencias de la imposibilidad de realizarla
  2. Desaparece la razón que causó el compromiso con la intención

Las intenciones condicionan el RP subsiguiente

- Adoptar una intención restringe el razonamiento práctico posterior
- Intenciones futuras no deben entrar en conflicto con las vigentes
  - Necesario un mecanismo de filtrado (i.e. filtro de admisibilidad [13])

Las intenciones condicionan las creencias futuras

Piénsese en un agente que adopta una intención concreta. Resulta obvio que entonces, asumiendo que ciertas condiciones básicas se cumplen, el agente cree que llegará a llevarla a cabo.

No es racional el adoptar una intención determinada si se tiene la creencia de que la situación ambiental derivada de ejecutar la intención es imposible.

Para que un agente adopte una intención deteminada, debe poseer cierta evidencia de que las condiciones ambientales que se persiguen son posibles.

El agente racional básico

```
Algoritmo:  Ciclo de control de agente, versión 1
1.  while true
2.          Observar el mundo
3.          Actualizar modelo del mundo
4.          Deliberar acerca de qué intención realizar
ahora
5.          Usar razonamiento medios-fines para obtener
plan
6.          Ejecutar el plan
7.  end while
```

Optimalidad en la racionalidad

La deliberación y el razonamiento medios-fines llevan un *coste de tiempo* asociado.

- El agente selecciona una acción en t_1 que era optimal en t_0
- El agente selecciona un plan en t_2 que era optimal en t_1

Optimalidad en la racionalidad (II)

El agente manifestará un comportamiento optimal cuando

1. los procesos deliberativo y de razonamiento medios-fines incurren en un tiempo suficientemente pequeño o
2. existe garantía de que el mundo va a permanecer estático en el intervalo $[t_0, t_2]$
3. cuando una intención es óptima en $[t_0, t_2]$

Solamente la primera va a ser alcanzable en el mundo real

Conclusión: el comportamiento del agente en un entorno real va a ser siempre aproximado

Alguna notación necesaria

- B se refiere a las creencias actuales del agente y *Bel* para su totalidad
- D será una variable para referirnos a los deseos y *Des* denotará el conjunto de todos los deseos.
- I representa un conjunto de intenciones e *Int* el conjunto de todas las intenciones posibles.
- Las percepciones a lo largo del tiempo serán $\rho, \rho', \rho_1, \ldots$ y *Per* el conjunto de todas las percepciones.
- Usaremos π para denotar un plan y *Plan* para el conjunto de todos los planes posibles.
  - $pre(\pi)$ es la precondición de π,
  - $post(\pi)$ es la postcondición y
  - $body(\pi)$ al cuerpo de π.

Alguna notación necesaria (II)

- *execute*(...) toma como entrada un plan y lo ejecuta hasta el final.
- $hd(\pi)$ es la cabeza del plan.
- Si el conjunto de acciones del plan es $\alpha_1, \alpha_2, \ldots, \alpha_n$ entonces $hd(\pi)$ es α_1.
- $tail(\pi)$ será la cola del plan, o sea $\alpha_2, \ldots, \alpha_n$.
- Sea π un plan, $I \subseteq Int$ un conjunto de intenciones y $B \subseteq Bel$ un conjunto de creencias.
- $sound(\pi, I, B)$ indica que π es un plan aceptable para obtener I dadas las creencias B.
- función de revisión de creencias: $brf : \varrho(Bel) \times Per \rightarrow \varrho(Bel)$
- proceso de deliberación: $deliberate : \varrho(Bel) \rightarrow \varrho(Int)$
- razonamiento medios-fines: $plan : \varrho(Bel) \times \varrho(Int) \rightarrow Plan$

El agente racional básico formalizado

```
Algoritmo:  Ciclo de control de agente, versión 2
1.  B := B_0;
2.  while true
3.          Obtener la percepción siguiente, ρ
4.          B := brf(B, ρ)
5.          I := deliberate(B)
6.          π := plan(B, I)
7.          execute(π)
8.  end while
```

Deliberación a fondo

Para que un agente delibere tiene que (1) generar el conjunto de acciones posibles y (2) escoger una acción para transformarla en intención
Descomponemos *deliberate*(B)

1. generación de opciones:

$$options : \varrho(Bel) \times \varrho(Int) \rightarrow \varrho(Des)$$

2. filtrado:

$$filter : \varrho(Bel) \times \varrho(Des)\varrho \times (Int) \rightarrow \varrho(Int)$$

Nueva versión del agente racional

```
Algoritmo:  Ciclo de control de agente, versión 3
1.  B := B_0;
2.  I := I_0;
3.  while true
4.              Obtener la percepción siguiente, ρ
5.              B := brf(B, ρ)
6.              D := options(B, I)
7.              I := filter(B, D, I)
8.              π := plan(B, I)
9.              execute(π)
10.  end while
```

Estrategias para el compromiso

¿Cuánto tiempo debería persistir una intención?, ¿bajo qué circunstancias debería una intención desaparecer como tal?
Estrategias para el compromiso [10]:

- Compromiso ciego: en donde el agente se compromete a realizar la intención y no la elimina hasta que verdaderamente la ha realizado (también se denomina compromiso fanático).
- Compromiso simple (*single minded commitment*): el agente se compromete a realizar la intención hasta que cree que bien la intención se ha realizado bien no es posible realizarla.
- Compromiso abierto (*open minded commitment*): un agente de este tipo va a mantener una intención tanto tiempo como esta sea posible, o dicho de otro modo, siempre que siga siendo un objetivo (i.e. un fin generado por el razonamiento medios-fines).

Compromiso a dos niveles

El agente anterior mantiene un compromiso con

- medios al no reconsiderar los planes
- fines al no reconsiderar sus intenciones

Reconsiderando el plan

```
Algoritmo:  Ciclo de control de agente, versión 4
1.  B := B_0;
2.  I := I_0;
3.  while true
4.              Obtener la percepción siguiente, ρ
5.              B := brf(B, ρ)
6.              D := options(B, I)
7.              I := filter(B, D, I)
8.              π := plan(B, I)
9.              while not empty(π) hacer
10.                     α := hd(π)
11.                     execute(α)
12.                     π := tail(π)
13.                     Obtener la percepción siguiente, ρ
14.                     B := brf(B, ρ)
15.                     si no sound(π, I, B) entonces
16.                             π := plan(B, I)
17.                     fin-si
18.             end while
19.  end while
```

Comentarios a la versión 4

- Reduce el compromiso con los medios del razonamiento medios-fines.
- Sigue manteniendo una estrategia de compromiso ciego.
- Solo reconsidera el plan (i.e. aumenta la reactividad)
- Diseño más realista

Reconsiderando los fines (i.e. intenciones)

Para ello introducimos una nueva función $succeeded(I, B)$

- true si se cumplen las intenciones I, dadas las creencias B, false si no

y la función $impossible(I, B)$

- true si dadas las creencias actuales B, las intenciones I no pueden cumplirse

Nueva versión

```
Algoritmo:  Ciclo de control de agente, versión 5
1.  B := B0;
2.  I := I0;
3.  while true
4.              Obtener la percepción siguiente, ρ
5.              B := brf(B, ρ)
6.              D := options(B, I)
7.              I := filter(B, D, I)
8.              π := plan(B, I)
9.              while not (empty(π) o succeeded(I, B) o impossible(I, B)) hacer
10.                     α := hd(π)
11.                     execute(α)
12.                     π := tail(π)
13.                     Obtener la percepción siguiente, ρ
14.                     B := brf(B, ρ)
15.                     si no sound(π, I, B) entonces
16.                             π := plan(B, I)
17.                     fin-si
18.             end while
19.  end while
```

Reconsiderando las intenciones (II)

Sofia es un agente BDI encargado de obtener documentos electrónicos. Un dia, el usuario le ordena que obtenga una copia electrónica de la tesis de U. N. Cualquiera y Sofia crea una intención para ello. La creencia de Sofia es que la tesis está en la página Web de U. N. Cualquier y crea entonces una intención para obtenerla de allí. Mientras que está realizando la planificación para la descarga del documento, el instructor le indica que existe una copia en local. Es considerar esta información haría mucho más eficiente la tarea de obtener la tesis al eliminar la descarga remota. Sin embargo, Sofía mantiene la intención de descargarlo en remoto ya que piensa que si lo hace habrá satisfecho su intención.

Observaciones

- Los pasos del 13 al 17 se reconsidera el plan **pero no las intenciones**.
- Si Sofía reconsiderara intenciones y plan, habría aprovechado la existencia de una copia en local del documento buscado.

Versión 6 del agente racional

```
Algoritmo:  Ciclo de control de agente, versión 6
1.  B := B_0;
2.  I := I_0;
3.  while true
4.              Obtener la percepción siguiente, ρ
5.              B := brf(B, ρ)
6.              D := options(B, I)
7.              I := filter(B, D, I)
8.              π := plan(B, I)
9.              while not (empty(π) o succeeded(I, B) o impossible(I, B)) hacer
10.                     α := hd(π)
11.                     execute(α)
12.                     π := tail(π)
13.                     Obtener la percepción siguiente, ρ
14.                     B := brf(B, ρ)
15.                     D := options(B, I)
16.                     I := filter(B, D, I)
17.                     si no sound(π, I, B) entonces
18.                             π := plan(B, I)
19.                     fin-si
20.             end while
21.  end while
```

Problema!

- Se han incluido en el ciclo de control todo el razonamiento práctico!
- Por un lado un agente que no reconsidera sus intencies frecuentemente continuará intentando cumplirlas en situaciones en que esto sea imposible
- por el otro, si se reconsideran demasiado se pierde tiempo en cumplir aquellas que son satisfacibles

Debe habilitarse un *trade-off* entre grado de commitment y nivel de reconsideración $\longrightarrow$ función que decida cuando es aconsejable

Agentes reactivos

- No presentan representación explícita de conocimiento simbólico complejo.
- Se trata de ofrecer respuestas inmediatas a estímulos del entorno.
- Justificación

  > *"la complejidad que puede llegar a manifestar el comportamiento de un agente debería ser reflejo de la complejidad del entorno en el que se desenvuelve, en lugar de ser el reflejo de el diseño interno del agente" (Simon, 1981. [?])*

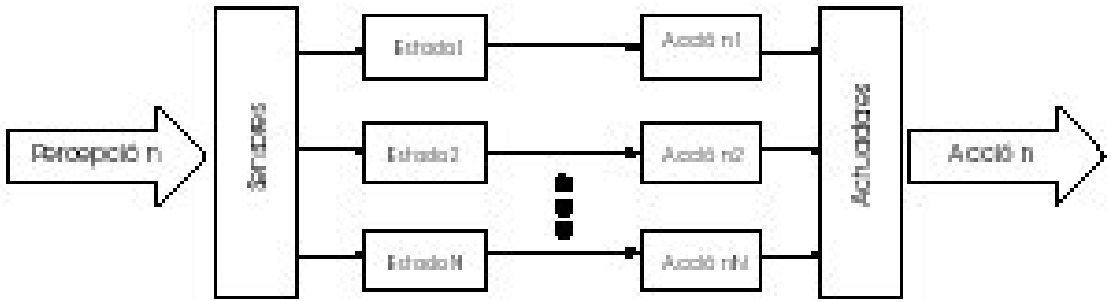
- Objetivo: comportamiento robusto en contraposición al *optimalmente correcto* en los BDI

Agentes reactivos (II)

- Denominados también agentes estímulo-respuesta [7]
- Generalmente, basados en sistemas de producción

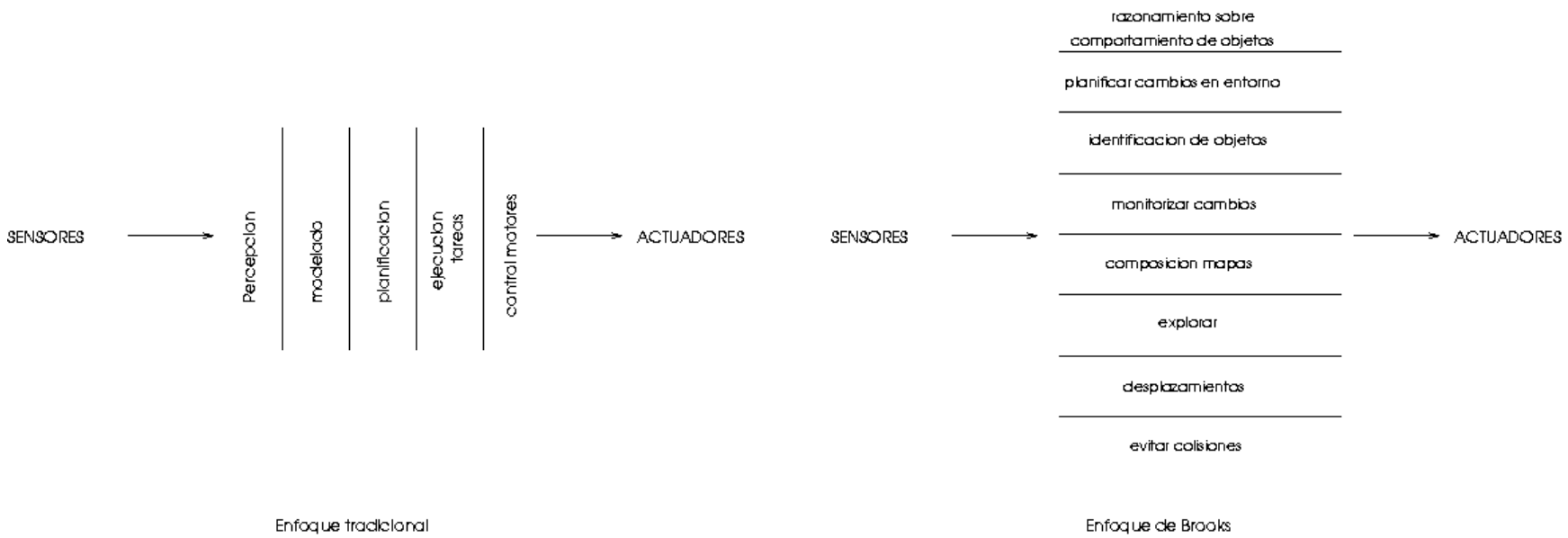
$$
\begin{aligned}
c_1 &\rightarrow a_1 \\
c_2 &\rightarrow a_2 \\
c_3 &\rightarrow a_3 \\
&\ldots \\
c_n &\rightarrow a_n
\end{aligned}
$$

- Otros casos, el de una red neuronal



Arquitecturas de subsumción

En lugar de desdoblar los sistemas inteligentes en niveles de funcionalidad se hace en niveles de competencia (Brooks, 1985 [2])

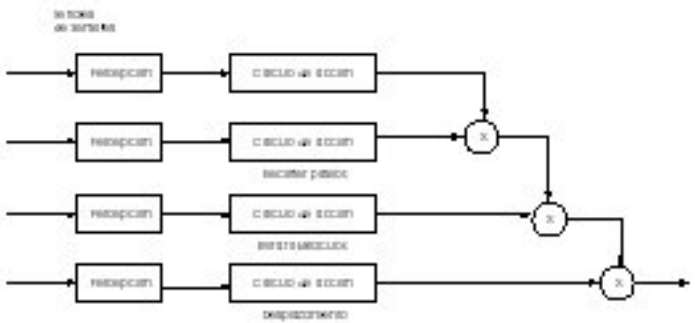


Máximas para diseño de robots con arquitecturas de subsunción

- El comportamiento complejo es un reflejo del entorno.
- Los diseños deben ser simples en términos de interfaces y complejidad de módulos
- El objetivo principal es el de construir robots barátos que pueden evolucionar en un espacio sin intervención humana de ninguna clase.
- El mundo del robot es tridimensional y no bidimensional.
- Los sistemas de coordenadas absolutos son una fuente de errores considerable. Los mapas relativos son más útiles para los robots móviles.
- Los robots no necesitan la construcción del entorno en base a modelos realistas a base de polihedros
- Los datos provenientes de sonar no son útiles para interacciones complejas aunque si para tareas básicas. Para taras más complejas ha de usarse visión artificial.
- El control de los sensores por parte del robot debe ser tal que permitan la recalibración sin manipulación externa (i.e. humana).
- Los tipos de robots a construir son criaturas artificiales y como tales, deben sobrevivir por dias, semanas, meses sin intervención humana. Por lo tanto, deben ser autosustentables.

Organización de módulos en arquitectura

- Cada nivel define una clase de comportamientos válidos
- el nivel n puede verse como una especialización del nivel $n-1$
- Operaciones entre módulos: supresión e inhibición
- Un nivel determinado es capaz de suprimir las entradas a un nivel inferior e inhibir, eliminando, las salidas de un nivel inferior

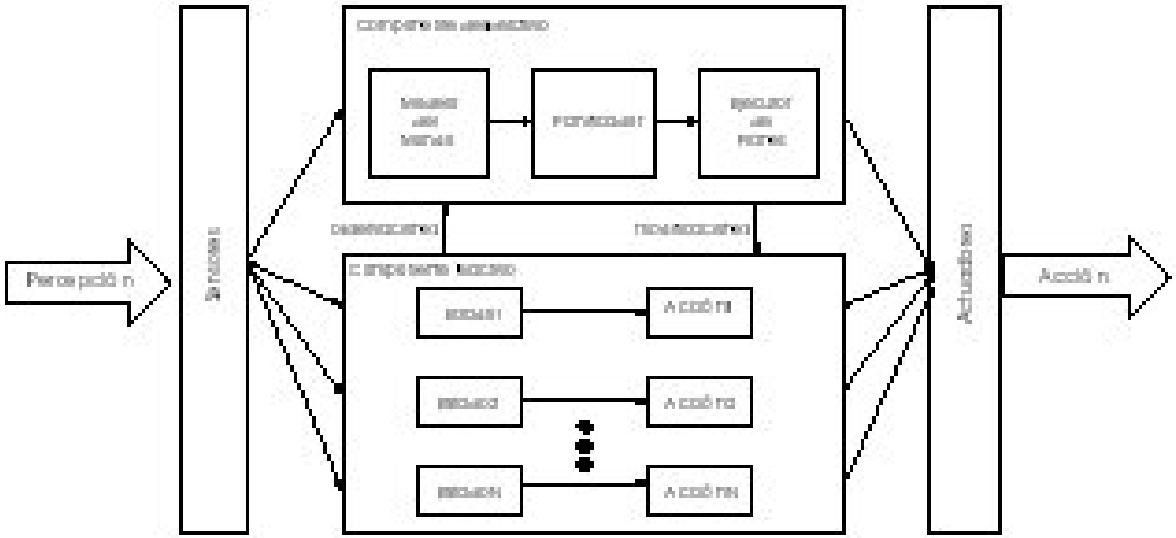


Construcción de robots incrementalmente

- Brooks plantea la construcción de agentes físicos incrementalmente
  1. Se comienza construyendo un robot que desarrolle compentencia de nivel 0 (i.e. que implemente completamente el nivel de competencia *evitar colisiones*).
  2. Se depura completamente el módulo
  3. Se construye un nuevo nivel de competencia (el nivel 1), capaz de
     - ★ leer datos de los interfaces de nivel 0
     - ★ inyectar datos a los mismos interfaces
- Interesante enfoque aunque no muy práctico (no es posible la construcción de robots complejos con tareas como planificación y cooperación)

Arquitecturas híbridas

Un agente híbrido típico dispone de componentes deliberativos que permiten realizar razonamientos complejos, realizar planes y tomar decisiones. Todo esto en combinación con componentes reactivos que permiten la reacción inmediata ante eventos para los cuales es necesario ese tipo de reacción.



Arquitecturas híbridas (II)

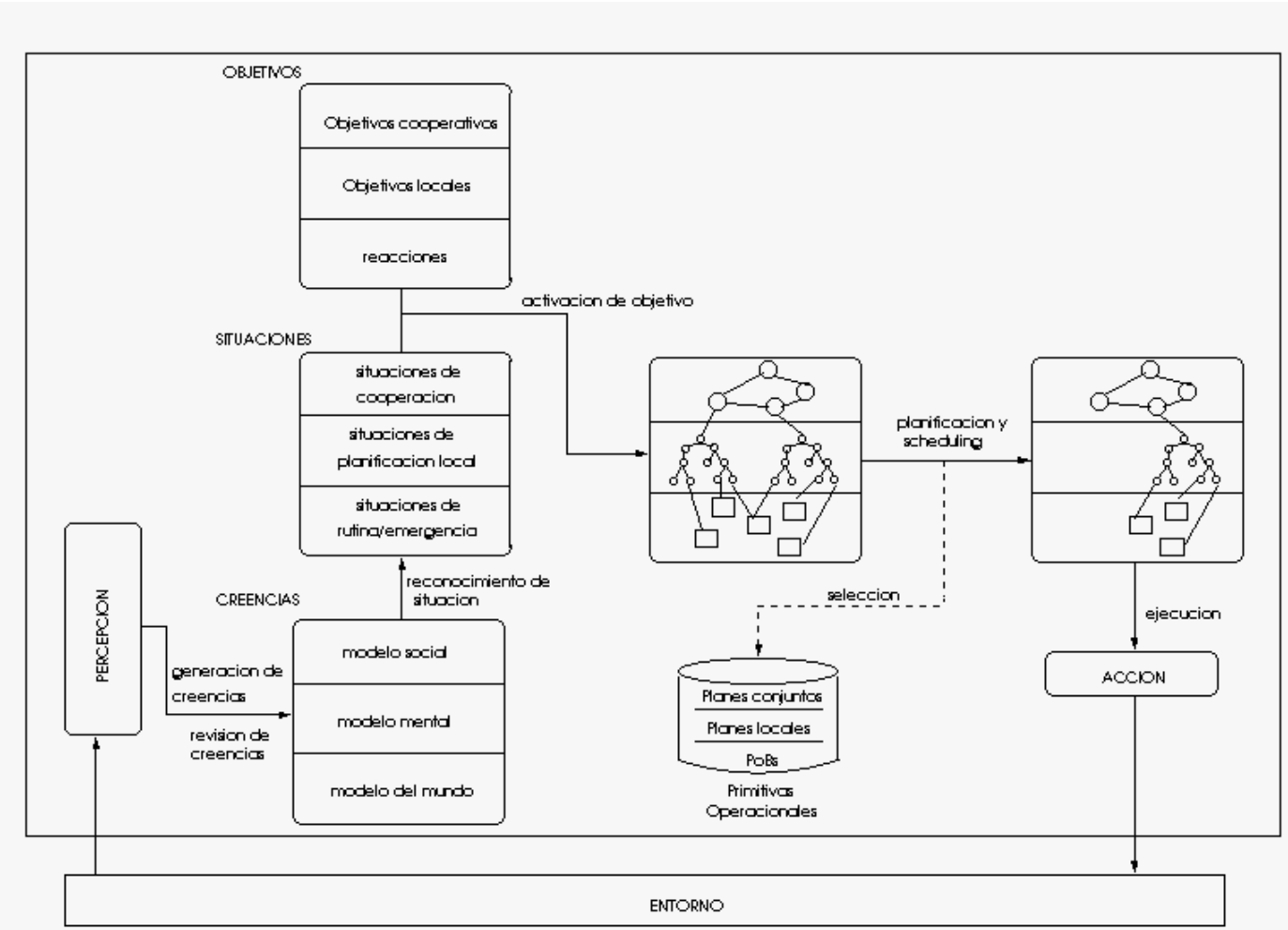
- Ejemplos: InteRRaP [5], Touring Machines [3] y PRS (*Procedural Reasoning System*) [4].
- En InteRRaP
  - Arquitectura en tres capas
    - ★ de comportamiento capacidad de decisión inmediata al trabajar con un modelo del mundo y estar en contato sensores y actuadores
    - ★ de planificación comportamiento deliberativo que maneja conocimiento necesario para planificación
    - ★ de cooperación da sociabilidad al agente y que maneja conocimiento de su entorno social

INTEgration of Reactive behaviour and Rational Planning (InteRRaP)

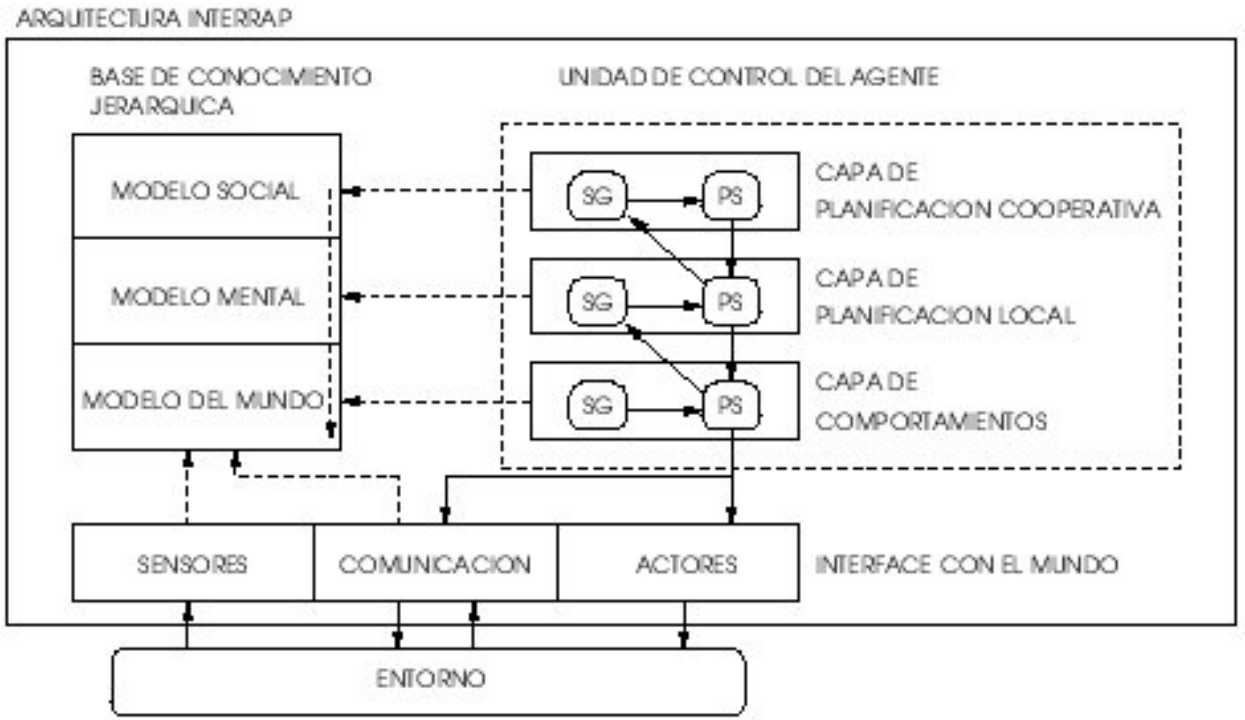
Requerimientos

- Comportamiento situado: de tal forma que los agentes reconozcan eventos inesperados y reaccionen, a tiempo y de forma apropiada, ante ellos.
- Comportamiento dirigido por objetivos: los agentes deben seleccionar las acciones a realizar en base a los fines que persiguen y teniendo en cuenta los medios disponibles para ello.
- Eficiencia: las tareas que surgen de la planificación han de ejecutarse de manera eficiente, atendiendo frecuentemente a restricciones de tiempo severas. Para ello se proporcionará un conjunto de procedimientos muy eficientes en su ejecución y con garantías en los resultados.
- Coordinación: cada agente debe tener en cuenta la presencia de otros agentes en el entorno y las interacciones tanto negativas como positivas que puedan tener lugar.

Modelo conceptual de INTERRAP



Arquitectura de INTERRAP



Conclusiones

- Los agentes son potentes herramientas para la construcción de software
- Su desarrollo no es dependiente de la tecnología
- Son una potente herramienta de modelado
- Existen distintos tipos de enfoques (reactivo, deliberativo e híbrido)


M. E. Bratman.
What is intention?
1990.


Rodney A. Brooks.
A robuts layered control system for a mobile robot.
Technical report, MIT, Artificial Intelligence Laboratory.
AI Memo 864.


I. A. Ferguson.
Touringmachines: Autonomous agents with attitudes.
IEEE Computer, 25(5):51–55, 1992.


M.P. Georgeff and A.L. Lansky.
Reactive reasoning and planning.
In *Readings in Plannings*, page 729734. 1990.


J. Muller and M. Pischel.
The Agent Architecture InteRRaP: Concept and application.
Technical Report RR-93-26, 1993.


BBC News.

Phone butler organises your life.
BBC News World Edition, July 2003.
http://news.bbc.co.uk/go/pr/fr/-/2/hi/technology/2996788.stm.


Nils J. Nilsson.
Artificial Intelligence: A New Synthesis.
Morgan Kaufmann, 1998.


Hyacinth S. Nwana.
Software Agents: An Overview.
Knowledge Engineering Review, 1996.


A. S. Rao and M. P. Georgeff.
BDI-agents: from theory to practice.
In *Proceedings of the First Intl. Conference on Multiagent Systems*, San Francisco, 1995.


Anand S. Rao and Michael P. Georgeff.
Modeling rational agents within a BDI-architecture.
In James Allen, Richard Fikes, and Erik Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of*

Knowledge Representation and Reasoning (KR'91), pages 473–484. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1991.


Yoav Shoham.
Agent-oriented programming.
Artficial Intelligence, 60(1):51–92, 1993.


M. Wooldridge and N. R. Jennings.
Intelligent agents: Theory and practice.
The Knowledge Engineering Review, 2(10):115–152, 1995.


Michael Wooldridge.
Reasoning about rational agents.
MIT Press, 2000.