



Arquitectura BDI

B	→	Beliefs
D	→	Desires
I	→	Intentions



Índice

- Introducción
- Visión intuitiva
- Teoría BDI
- Arquitectura BDI
- Un ejemplo de sistema: dMARS
- Un ejemplo de lenguaje: AGENTSPEAK(L)
- Conclusiones Finales



Introducción (I)

- En la arquitectura BDI el agente es visto como un agente racional con un conjunto de actitudes mentales.
 - Creencias, Deseos e Intenciones.
- El agente toma sus acciones en función de sus estados.
- Esta aproximación es considerada como muy apropiada para agentes en entornos reales complejos y dinámicos.



Introducción (II)

- Se han desarrollado trabajos para dotar de una semántica a los agentes BDI.
 - Cohen & Levesque, Lógica BDI (Rao & Georgeff)
- Se han implementado sistemas basados en agentes BDI para problemas reales
 - PRS, dMARS.
- Problema: existe un hueco entre el empleo de formalismos de una parte y el vocabulario informal de la otra.



Visión intuitiva (I)

- Elementos que dan nombre al modelo BDI:
 - Una **creencia** del agente sería información que tiene el agente sobre el entorno.
Componente informativo del agente
 - Un **deseo** del agente es algo que el agente quiere conseguir y que por tanto tiene preferencia.
Componente de motivación del agente



Visión intuitiva (II)

- Los **objetivos** serían un subconjunto de deseos consistentes entre sí que el agente debe perseguir.
- Las **intenciones** del agente son los objetivos elegidos. Existe un compromiso por parte del agente a conseguir sus intenciones.
Componente deliberativo del sistema



Teoría BDI

- Una teoría de agente puede ser considerada como una **especificación formal** de un *agente*.
- **Agente** → Entidad que aparece como el sujeto de creencias, deseos, etc.
 - ⇒ **Stma. Intencional** → Entidades cuya conducta puede ser predicha por el método de la atribución de creencias, deseos e ingenio racional.



Teoría BDI

Formalismo Rao & Georgeff (I)

- Basado en la lógica CTL*.
 - Dicha lógica es una lógica proposicional ramificada temporalmente.
- ↓
- extendida a un entorno de mundos posibles.
 - introducción de operadores modales para creencias, objetivos e intenciones.



Teoría BDI

Formalismo Rao & Georgeff (II)

- Mundos Posibles
- Representar el sistema mediante un árbol de decisión, donde:
 - los nodos representan estados y
 - los arcos los eventos.
- Hay dos tipos de nodos:
 - *chance nodes*: cambios entorno no controlados
 - *choice nodes*, acciones del sistema.



Teoría BDI

Formalismo Rao & Georgeff (III)

- Mundos Posibles
- El árbol se transforma a un modelo equivalente de conjuntos de mundos posibles.

algoritmo de transformación [Rao y Georgeff91]
- Este modelo consiste en un conjunto de mundos donde:
 - cada mundo es una estructura de árbol.
 - un índice de un mundo es una *situación*.
 - Para cada situación asociamos un conjunto de mundos de creencias, de deseos y de intenciones.



Teoría BDI

Formalismo Rao & Georgeff (IV)

- Lógica BDI (Semántica)
 - Estructura temporal: tiempo ramificado en el futuro y un pasado único.
 - Una situación es un punto de tiempo en un mundo determinado.
 - Los eventos transforman un punto de tiempo en otro: *primitivos*, *no primitivos*.



Teoría BDI

Formalismo Rao & Georgeff (V)

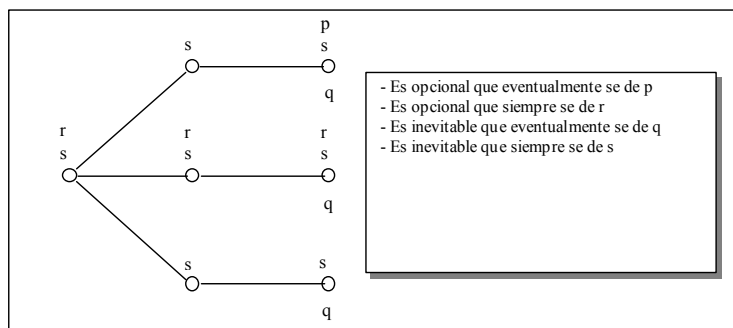
- Lógica BDI (Semántica)
 - Las ramas indican las opciones en un momento dado.
 - Un evento ejecutado puede tener o no éxito.
 - Se distingue entre:
 - *state formulas* (evaluadas en una situación)
 - *path formulas* (evaluadas en una ruta)

Teoría BDI Formalismo Rao & Georgeff (VI)

- Lógica BDI (Semántica)
 - Hay dos operadores modales: *opcional* e *inevitable*, operan sobre "path formulas".
 - Una "path formula" es opcional si es verdadera en al menos una ruta.
 - Una "path formula" es inevitable si es verdadera en todas las rutas.
 - Operadores temporales standard:
 - ◇ eventually □ always ∪ until ○ next

Teoría BDI Formalismo Rao & Georgeff (VII)

- Lógica BDI (ejemplo)

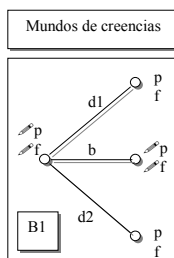


Teoría BDI Formalismo Rao & Georgeff (VII)

- Lógica BDI
- **BELIEFS:** a cada situación se le asocia un conjunto de "*beliefs-accessible worlds*" (mundos que el agente cree posibles).
- **GOALS:** cada situación dispone del conjunto de "*goals-accessible worlds*" para representar los objetivos del agente. *Goals* son deseos elegidos que sean consistentes.
- **INTENTIONS:** representados por "*intention-accessible worlds*", son mundos que el agente se ha comprometido a intentar alcanzar.

Teoría BDI Formalismo Rao & Georgeff (VIII)

- Lógica BDI



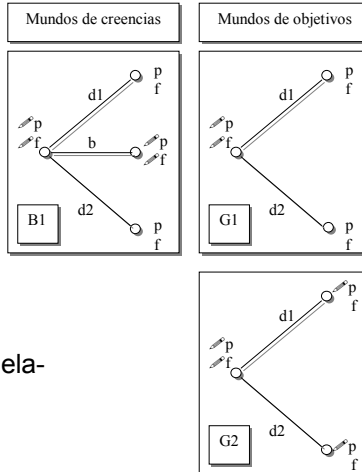
- f -sacarte una muela-
- p -tener dolor-



Teoría BDI

Formalismo Rao & Georgeff (VIII)

■ Lógica BDI



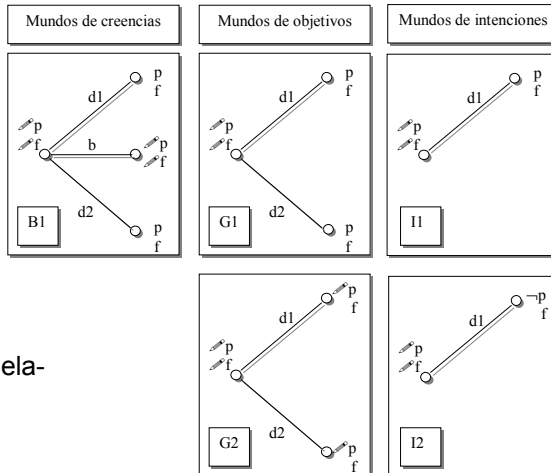
- f -sacarte una muela-
- p -tener dolor-



Teoría BDI

Formalismo Rao & Georgeff (VIII)

■ Lógica BDI



- f -sacarte una muela-
- p -tener dolor-



Arquitectura BDI

■ Intérprete básico

- Este intérprete implementa las entidades definidas por los operadores modales como estructuras de datos :

```
BDI-Interprete  
Inicializar-estado();  
Repetir  
    Opciones := generador-opcion(cola-eventos, S);  
    Opciones-seleccionadas := deliberar(opciones, S);  
    Modificar-intenciones(Opciones-seleccionadas, S);  
    Ejecutar(S);  
    Cola-eventos := Tomar-nuevos-eventos-externos();  
Fin repetir
```



Arquitectura BDI

■ Arquitectura abstracta

- Consideramos en este punto el caso donde se ha refinado tanto el estado del sistema como el intérprete
- Comprende 3 estructuras de datos globales,
 - una para cada actitud, más una cola para gestión de eventos.
- Se permiten operaciones de modificación y consulta
- No se da información acerca de cómo representar las estructuras necesarias



Arquitectura BDI

■ Arquitectura abstracta

- Un intérprete, refinado respecto al anterior, ejecuta el siguiente bucle:

```
BDI-Interpreter  
Inicializar-estado();  
Repetir  
  Opciones := generador-opcion(cola-eventos);  
  Opciones-seleccionadas := deliberar(opciones);  
  Modificar-intenciones(Opciones-seleccionadas);  
  Ejecutar();  
  Tomar-nuevos-eventos-externos();  
  Borrar-actitudes-con-éxito();  
  Borrar-actitudes-imposibles();  
Fin repetir
```



Arquitectura BDI

■ Arquitectura abstracta

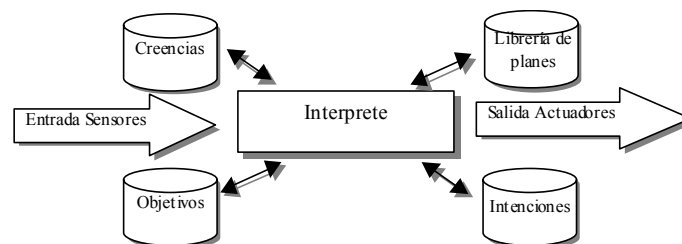
- Esta arquitectura es una idealización que captura la teoría vista. **No practicable.**
 - Los procedimientos de probabilidad requeridos no son computables.
 - No se han dado indicaciones de cómo el generador de opciones y el procedimiento de deliberación pueden ejecutarse lo suficientemente rápidos para satisfacer demandas de tiempo real.

Arquitectura BDI

- Arquitectura abstracta
 - La idea es desarrollar una simplificación (simplificación de PRS).
 - Sólo creencias del estado actual.
 - Representan el conocimiento como planes.
 - Evento disparo; Condiciones; Cuerpo
 - Una intención es representada usando una pila de planes relacionados jerárquicamente.
 - Pueden coexistir múltiples pilas.

Arquitectura BDI

- Implementación
 - Existe una típica arquitectura adoptada por las diferentes implementaciones de sistemas basados en BDI





Arquitectura BDI

■ Implementación

- El bucle principal del intérprete es similar al visto,
- Los procedimientos que aparecen son más rápidos para satisfacer las demandas de aplicaciones reales



Arquitectura BDI

■ Implementación

- *generador de opciones*:
 - itera sobre la biblioteca de planes y retorna aquellos que cumplan la condición de invocación (hace matching con el evento) y cuyas precondiciones son creídas por el agente.

```
generador_de_opciones( eventos_de_disparo )
opciones := { };
para cada evento_disparo hacer
  para cada plan hacer
    si matching( invocación(plan), evento_disparo) entonces
      si probable(precondición(plan), B) entonces
        opciones := opciones U {plan};
retornar(opciones)
```



Arquitectura BDI

■ Implementación

■ *deliberación:*

- su tiempo de ejecución depende de las restricciones del entorno, una selección aleatoria podría ser suficiente. En otros casos la inclusión de metaplanes puede ser útil.

```
deliberar(opciones)
si longitud(opciones) <= 1 entonces retornar(opciones);
sino opciones-metanivel
    := generador-opciones(b-add(conjunto-opciones(opciones)));
opciones-seleccionadas := deliberate(opciones-metanivel);
si nulo(opciones-seleccionadas) entonces
    retornar(elección-aleatoria(opciones));
sino retornar(opciones-seleccionadas)
```



Arquitectura BDI

■ Implementación. Bucle intérprete:

- Observa el mundo y el estado interno.
- Genera nuevos posibles deseos, encontrando planes que pueden ser disparados.
- Selecciona de ese conjunto de planes posibles a disparar a uno para su ejecución. (plan instanciado)
- Dicho plan instanciado puede ser disparado por una intención y se inserta en la pila de dicha intención o bien se crea una nueva intención.
- Se selecciona la intención a ejecutar del conjunto de intenciones.



dMARS (I)

- dMARS

(*distributed Multi-Agent Reasoning System*)

- Sucesor de PRS, basado en el modelo BDI.
 - *Sistema basado en agentes que más ha sido empleado desde el punto de vista práctico.*
- Partimos de la especificación formal realizada entre otros por Mark d'Inverno.
- Representa el modelo operacional del sistema mediante Planes.
- Monitoriza el mundo y su propio estado, generando eventos almacenados en una cola.



dMARS (II)

- Elementos que constituyen el sistema:

- **Creencias**, características:

- Son literales de la lógica de 1º orden.
- Una creencia se expresa como una fórmula donde el número de variables es 0.

- **Objetivos**, pueden ser de dos tipos:

- *achieve*: representado por una creencia precedida del operador !
- *query*: representado por una expresión cuya verdad se evalúa frente al conjunto de creencias precedida por el operador ?



dMARS (III)

■ **Acciones**, quedan representadas por:

- **internas**: las cuales pueden añadir o borrar creencias.
- **externas**: son llamadas a procedimientos
 - formadas por un *símbolo de acción externa* y
 - un *conjunto de términos* o parámetros.



dMARS (IV)

■ **Planes**, cada plan en dMARS consta de 6 elementos:

- *condición de invocación*: representa el evento de disparo del plan.
- un *contexto opcional*: donde incorporar las posibles precondiciones a cumplirse previas a la ejecución del plan.
- un *cuerpo*: un grafo donde los arcos se etiquetan con objetivos o acciones y los estados en si carecen de importancia
- una *condición de mantenimiento*: debe ser cierta para poder continuar con el plan.
- un *conjunto de acciones internas*: a ejecutar si el plan falla.
- otro *conjunto de acciones internas*: a ejecutar si el plan tiene éxito.



dMARS (V)

- **Planes**, distinguimos entre:

- *Plan relevante*: es aquel plan para el que se cumple su condición de disparo.
- *Plan aplicable*: es un plan relevante para el que también se cumple su contexto, es decir su contexto es una consecuencia lógica de las creencias del agente.



dMARS (VI)

- **Intenciones**

- Las intenciones son secuencias ordenadas de instancias de plan.
 - ➡ **planes que se han activado**
- Una intención es como una pila de instancias de plan.
- Se crea como respuesta a un evento externo,
 - Inicialmente consta de una instancia de plan.
 - La llegada de eventos generados por la propia intención provocarán nuevas instancias de plan a concatenar.



dMARS (VII)

- **Eventos**, un evento consta de:
 - un evento de disparo.
 - una identificación de instancia de plan que creó el evento (sólo en el caso de ser un evento interno).
 - un conjunto de instancias de plan que pueden haber fallado.
 - un entorno.
- Podemos distinguir entre:
 - *Evento externo*: su llegada provoca la generación de una instancia de plan para su resolución.
 - *Evento interno*: ocurre cuando la rama a ejecutar de la intención activa es un objetivo a alcanzar y que no puede ser alcanzado inmediatamente.



dMARS (VIII)


- **Agente dMARS**, consta de:
 - *biblioteca de planes.*
 - *función de selección de intenciones.*
 - *función de selección de planes.*
 - *función de selección de eventos.*
 - *función de selección de sustituciones.*
 - *función para determinar la siguiente rama de un plan.*
- El estado de un agente es aquello que varía con el tiempo:
 - **Creencias del agente.**
 - **Intenciones del agente.**
 - **Cola de eventos para ser procesados.**
- Inicialmente el agente es dotado de unas creencias e intenciones y su percepción le permite obtener eventos externos.



dMARS (IX)

Ejecución Interprete


- El intérprete realiza un proceso cíclico:
 - 1. Monitorización del mundo y del estado interno actualizando la cola de eventos.
 - 2. Generación de nuevos deseos viendo que planes pueden ser disparados de acuerdo a los nuevos eventos.
 - 3. Seleccionar del conjunto de nuevos planes uno de ellos. Se genera un plan instanciado.
 - 4. Colocar el nuevo plan instanciado como una nueva intención o en la pila de una intención anterior.
 - 5. Seleccionar una intención del conjunto de intenciones activas y ejecutar un nuevo paso del plan situado en la cabeza de la pila de la intención.
 - Si el paso a realizar es un acción entonces ejecutarla.
 - Si el paso es un subobjetivo, entonces se genera un evento interno.



dMARS (X)

Procesado de eventos

- El procesado de eventos es como sigue:
 1. Si hay eventos en la cola se selecciona el 1º (mediante la función de selección)
 2. Se determina el conjunto de planes aplicables y se selecciona uno solo para crear la instancia de plan correspondiente.
 3. Si el evento es externo se creará una nueva intención. Por otra parte, si el evento es interno la instancia del plan se coloca en la pila de la intención que generó el evento.



dMARS (XI)

Ejecución de intenciones

- La ejecución de una intención es como sigue:
 1. El primer paso consiste en seleccionar la intención a ejecutar (función de selección de intenciones).
 2. Dentro de la intención seleccionada se identifica la instancia de plan a ejecutar, (cabeza de la pila de la intención).
 3. Se selecciona la rama de la instancia de plan seleccionada. Se pasa a ejecutar dicha rama.



dMARS (XII)

Ejecución de intenciones

- El movimiento entre estados de una instancia se realiza según el resultado de la ejecución de la rama seleccionada.
 - Si la rama tiene éxito se pasa al siguiente estado apuntado por la rama.
 - Si falla habrá que borrar dicha rama e intentar otra si es posible.



dMARS (XIII)

Ejecución de intenciones

- La ejecución de una rama puede llevar a 4 casos (función del contenido de dicha rama):
 - Es una acción **externa**, se ejecuta inmediatamente.
 - Si la rama tiene éxito se pasa al siguiente estado de la instancia.
 - Si la rama falla se borra dicha rama
 - Es una acción **interna**, la base de datos debe ser modificada de acuerdo a la acción. Una acción interna siempre es exitosa.
 - Es un objetivo tipo **pregunta**, lo que se hace es ver si el entorno aplicado al objetivo puede ser unificado con el conjunto de creencias.
 - Es un objetivo tipo **alcanzar**, si el objetivo puede ser unificado con las creencias el resto del plan es unificado y la rama tiene éxito.
- Posteriormente, el ciclo global volvería a iniciarse.



dMARS (XIV)

Ejecución de intenciones

- El éxito de un plan se produce cuando una rama que ha tenido éxito en su ejecución llega a un estado final.
- Si por el contrario se llega a un estado con más ramificaciones se pasaría a intentar ejecutar una rama del nuevo estado.
- A partir de que el plan haya tenido éxito nos podemos encontrar con dos casos:
 - Que queden más planes en la intención del plan finalizado.
 - No quedan más planes en la intención del plan finalizado:
 - entonces la intención ha tenido éxito y se elimina.



Conclusiones Finales

- Numerosos artículos han ido apareciendo sobre agentes BDI.
- Podemos ver dos líneas:
 1. Para la **formalización** de estas nociones se ha empleado diferentes lógicas.
 2. **Implementaciones** de agentes BDI, se han realizado asumiendo simplificaciones.
 - Las actitudes mentales son estructuras de datos
 - El empleo de planes aumenta la velocidad
- Debido a las simplificaciones las implementaciones carecen de base teórica.



Conclusiones Finales

- Críticas sobre los agentes BDI:
 - El *porqué* de dichas tres actitudes (creencias, deseos e intenciones) → compromisos, capacidades, obligaciones, elección.
 - La *utilidad* de las lógicas BDI multi-modal se ha puesto en entredicho, → poca relevancia desde el punto de vista práctico.



Referencias

- [Bratman87] Bratman, M. E., Israel, D. J., Pollack, M. E.: Toward an architecture for resource-bounded agents. *Technical report CSLI-87-104*. Center for the Study of Language and Informations, SRI and Stanford University, 1987.
- [Rao and Georgeff95] Georgeff, M., Rao A. BDI Agents: From Theory to Practice. En Proceedings of the First International Conference on Multi-Agents Systems (ICMAS-95), San Francisco, San Mateo, USA, 1995.
- [Rao and Georgeff91] Georgeff, M., Rao A. Modeling Rational Agents within a BDI-Architecture. En Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann, San Mateo, CA, 1991.
- [Müller96] Müller, J. The Design of Intelligent Agents: A Layered Approach. LNAI Vol. 1177, Springer Verlag, 1996
- [Emerson90] Emerson, E. Temporal and Modal Logic. En J. Vab Leeuwen, editor, Hanbook of Theoretical Computer Science, volume B. Elsevier Science, 1990.
- [Wooldridge95] Wooldridge, M.; Jennings, N. R.; "Intelligent Agents: Theory and Practice", The Knowledge Engineering Review, vol. 10 (2) pp. 115-152, 1995.
- [d'Inverno97] d'Inverno, M. Kinny, D. Luck, M. Wooldridge, M. A Formal Specification of dMARS. Fourth International Workshop on Agent Theories, Architectures, and Languages, ATAL'97, 1997
- [Dennet87] Dennet, D.C. The Intentional Stance. The MIT Press, Cambridge, MA, 1987.