

How To Do The Right Thing

Pattie Maes

MIT. Artificial Intelligence Laboratory

1989

Definición del problema

- Agente autónomo (Ej: robot en Marte)
- Debe cumplir una serie de objetivos
- Entorno dinámico y complejo
- Recursos computacionales y temporales limitados

¿Qué acción realizamos en cada momento?

Planteamiento

- Agente autónomo: colección de módulos en competencia
- ¿Cómo se seleccionan los módulos?:
Activación/Inhibición
- Reactivo y deliberativo
- Seleccionar acciones *suficientemente buenas*:
 - *Orientadas a un objetivo*
 - *Adaptable al entorno actual*
 - *Favorecen el objetivo/plan actual*
 - *Hace “planes” para evitar situaciones no deseadas*
 - *Robusto, reactivo y rápido*
- Distribuido: no modulos que elijen qué acción se realiza

Algoritmo

- Cada módulo se define como una tupla $(c_i, a_i, d_i, \alpha_i)$:
 - c_i Lista de precondiciones
 - a_i Add list
 - d_i Delete list
 - α_i Nivel de activación
- Un módulo es ejecutable en t cuando se cumplen sus precondiciones

Algoritmo

- **Activación/Inhibición**

- Los módulos están unidos por tres tipos de enlaces:

$$x = (c_x, a_x, d_x, \alpha_x) \quad y = (c_y, a_y, d_y, \alpha_y)$$

- Enlaces sucesores (activan los módulos ejecutables):

- x tiene a y como sucesor: $p \in a_x \cap c_y$

- Enlaces predecesores (activan los módulos no ejecutables)

- x tiene a y como predecesor: $p \in c_x \cap a_y$

- Enlaces de conflicto (inhiben todos los módulos)

- y tiene un conflicto con x : $p \in c_x \cap d_y$

Algoritmo

- **Activación/Inhibición**
- Influencia del entorno y objetivos:
 - Activación por el estado: módulos que cumplen parcialmente el estado
 - Activación por objetivos: módulos que logran un objetivo
 - Inhibición por objetivos protegidos: módulos que eliminan objetivos ya logrados

Algoritmo

- El algoritmo es un bucle que hace lo siguiente en cada interacción sobre cada módulo:
 1. Se calcula la activación/inhibición del estado, objetivos y objetivos protegidos
 2. Se calcula la activación/inhibición de los enlaces sucesores, predecesores y conflictos
 3. Función que asegura que los niveles de activación son constantes ¿?
 4. El módulo que cumple estos tres objetivos se activa:
 1. Es ejecutable
 2. El nivel de activación supera un determinado límite
 3. Si dos módulos cumplen lo anterior, se eligen aleatoriamente

Ejemplo

- Robot con dos “manos”, cuyo objetivo es pintarse a si mismo y lijar un tablero.
- Cuando se pinta a si mismo queda no operativo

STATE OF THE ENVIRONMENT: (HAND-IS-EMPTY HAND-IS-EMPTY
SANDER-SOMEWHERE SPRAYER-SOMEWHERE
OPERATIONAL BOARD-SOMEWHERE)

GOALS OF THE ENVIRONMENT: (BOARD-SANDED SELF-PAINTED)

PROTECTED GOALS OF THE ENVIRONMENT: NIL

Ejemplo

- Definición de algunos módulos:

```
(defmodule PICK-UP-SPRAYER
```

```
  : condition-list '(sprayer-somewhere hand-is-empty)
```

```
  : add-list       '(sprayer-in-hand)
```

```
  : delete-list    '(sprayer-somewhere hand-is-empty))
```

```
(defmodule PUT-DOWN-SANDER
```

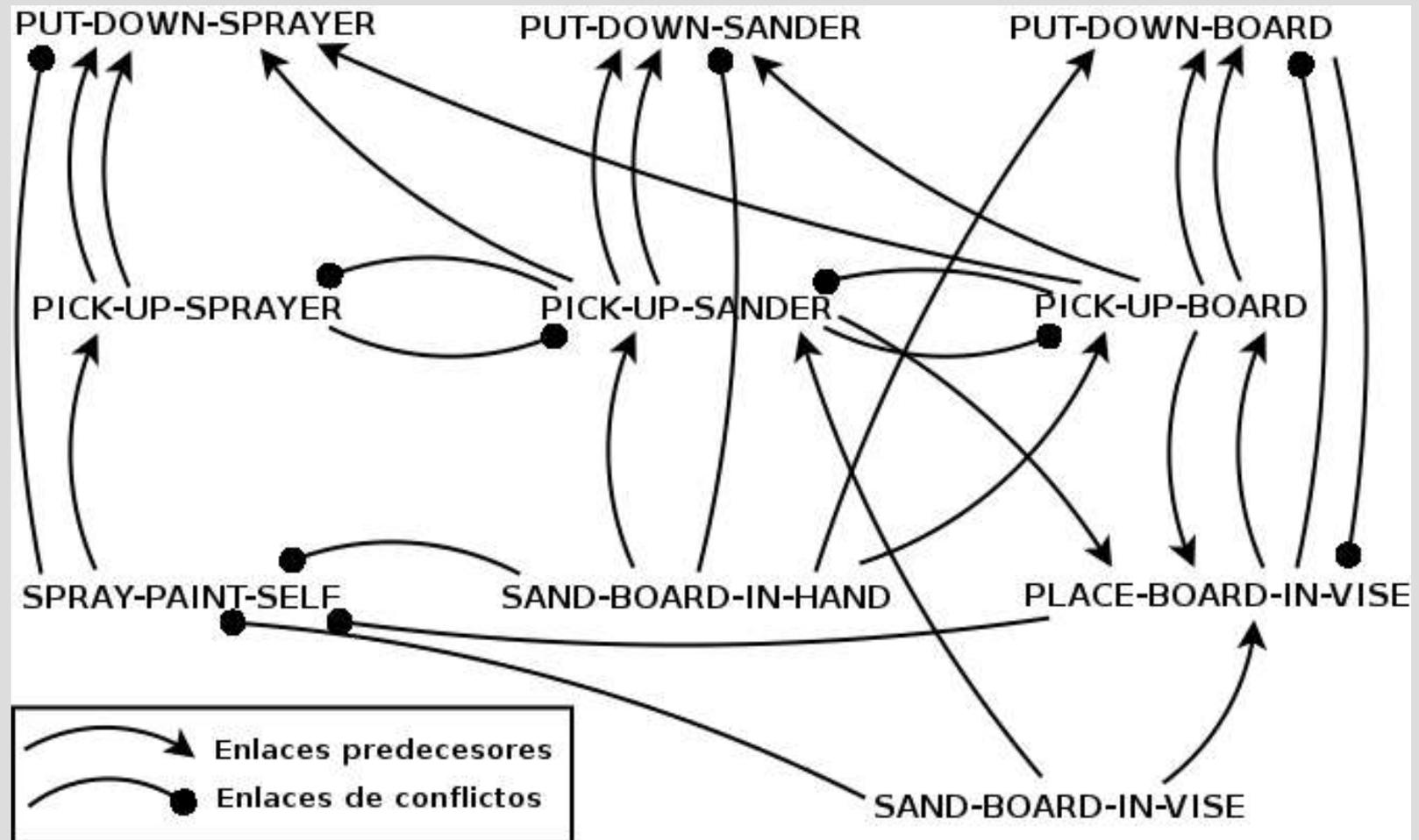
```
  : condition-list '(sander-in-hand)
```

```
  : add-list       '(sander-somewhere hand-is-empty)
```

```
  : delete-list    '(sander-in-hand))
```

Ejemplo

- Difusión de la energía de activación:



Resultados

- Gracias a los parámetros globales:
 - Orientado a objetivos
 - Relevancia de la situación actual
 - Adaptable a los cambios en el entorno
 - Tendencia hacia los planes/objetivos actuales
 - Evitar conflictos de objetivos
 - Rapidez

Limitaciones

- Lenguaje de descripción muy simplificado.
No hay variables
- No hay historia, por tanto puede haber bucles
- ¿Cómo seleccionar los parámetros globales?