

Robótica Basada en Comportamientos

Comportamiento Robótico

Facultad de Ingeniería
Instituto de Computación

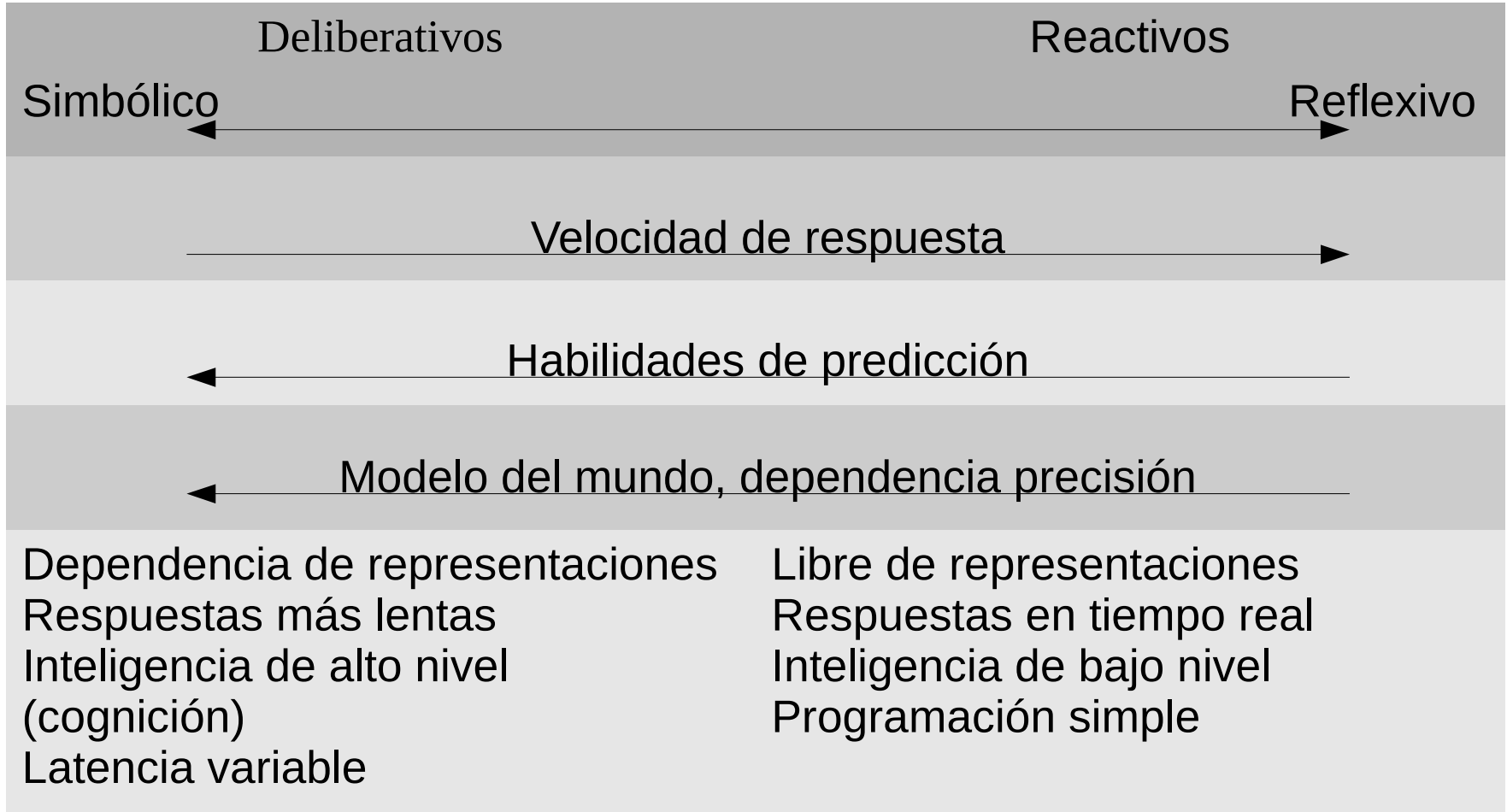
Objetivos

- Entender
 - qué son comportamientos robóticos.
 - métodos para expresar y codificar comportamientos.
 - Cómo componer y coordinar múltiples comportamientos.
- Decisiones de diseño.

Definiciones

- Comportamientos como respuestas a estímulos.
- Sistemas robóticos puramente reactivos.
 - Percepción y acción fuertemente acoplados.
Comportamiento como bloque básico compuesto de una pareja sensorial-motora.
 - No se utilizan representaciones abstractas.
WYSIWYG
 - Bioinspirados.
A menudo los modelos de comportamiento animal sirven como inspiración.
 - No se revisa el pasado.
 - Modulares desde la perspectiva del diseño del software.

Sistemas Robóticos : Espectro



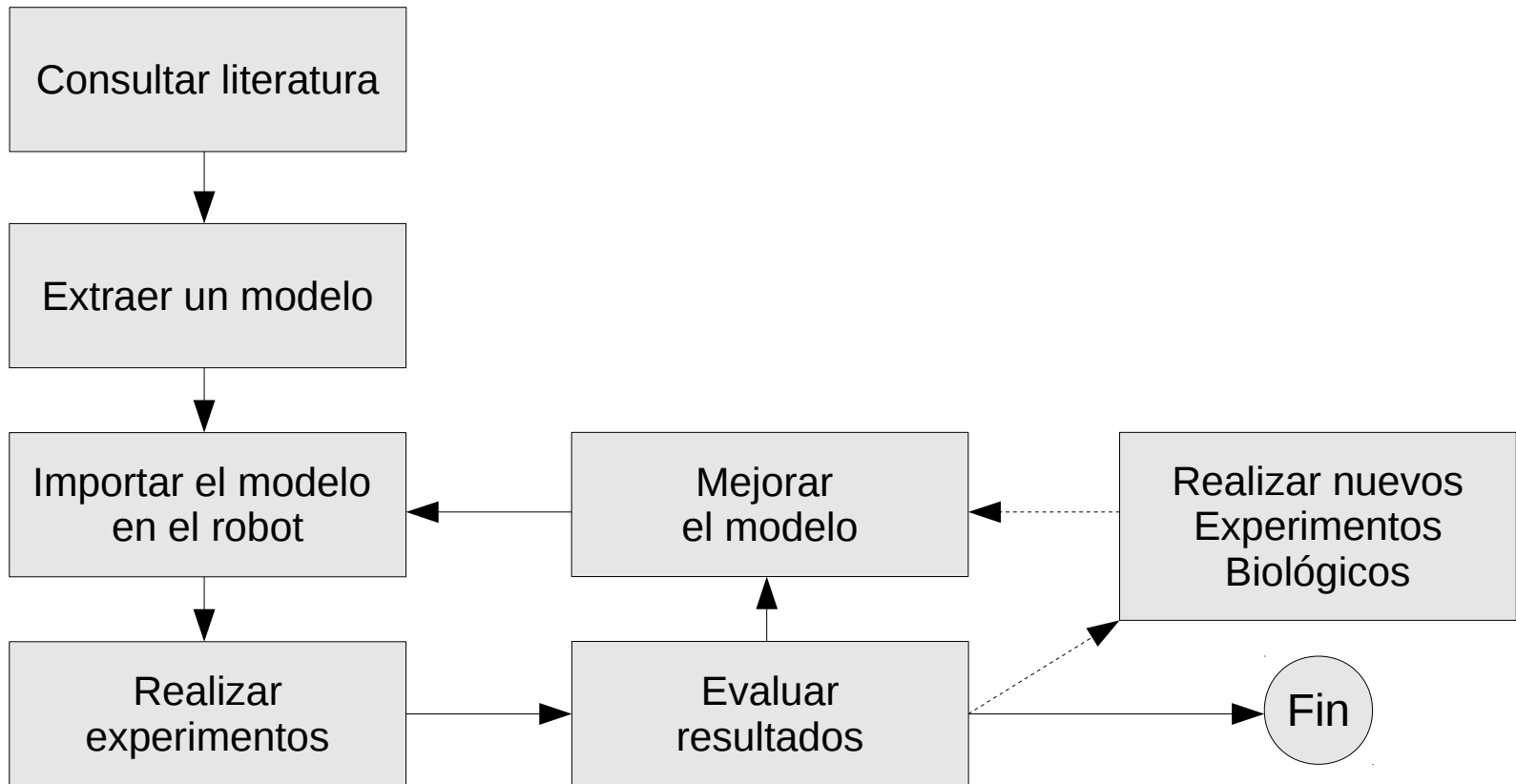
Bases del comportamiento robótico

- ¿De dónde provienen los comportamientos robóticos?
 - ¿Cuáles son los comportamientos básicos correctos para un sistema robótico?
 - ¿Qué es un comportamiento básico o primitivo?
 - ¿Cómo coordinar eficientemente esos comportamientos?
 - ¿Cómo se relacionan con sensores y actuadores en la práctica?

Especificación y Diseño ^{1/3}

- Diseño guiado/restringido por la etología.
 - Guiado
 - Clasificaciones
 - Composiciones
 - Especificaciones
 - Restringido
 - Emular
 - Modelos predictivos

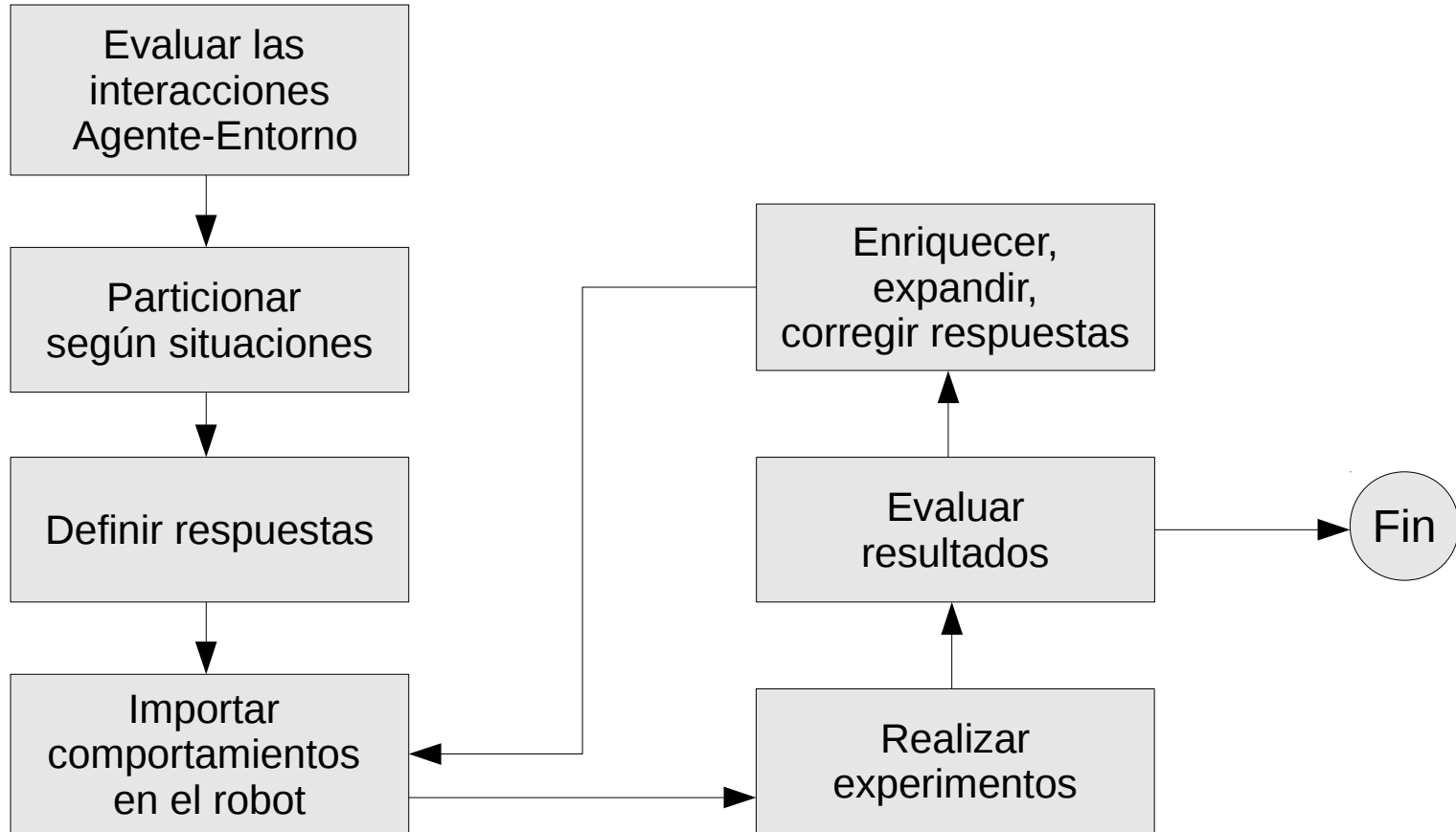
Diseño guiado por Etología



Especificación y Diseño ^{2/3}

- Diseño basado en situaciones.
 - La elección de la acción está relacionada íntimamente con la situación en la que se encuentra el robot.
 - La percepción se reduce a reconocer situaciones.
 - Requiere sólidos conocimientos sobre la relación entre el agente y su entorno.

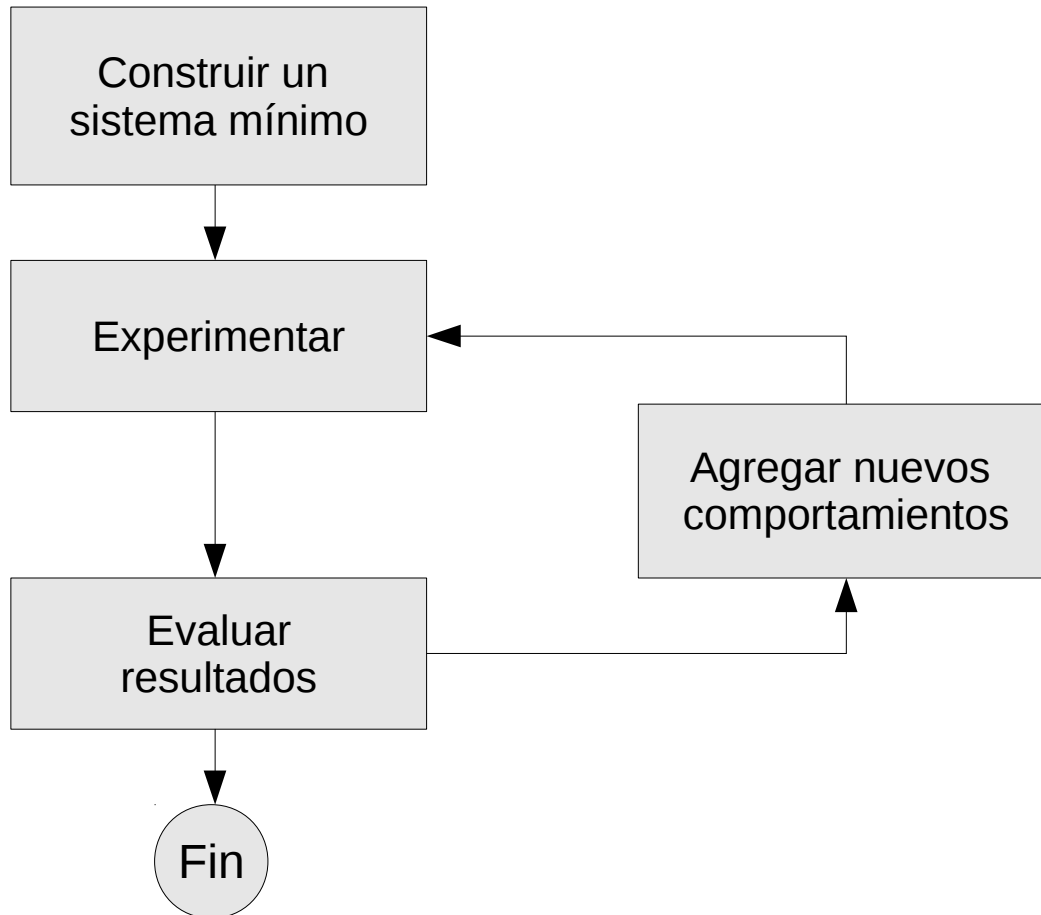
Diseño basado en situaciones



Especificación y Diseño ^{3/3}

- Diseño definido experimentalmente.
 - Los comportamientos son creados de forma bottom-up, guiados por la experimentación.
 - La premisa es:
 - dotar al robot de un conjunto limitado de habilidades o competencias.
 - experimentar en el entorno.
 - determinar qué funciona y qué no.
 - corregir comportamientos defectuosos.
 - agregar nuevos comportamientos.

Diseño definido experimentalmente



Categorización ^{1/4}

- **Exploración/Direccionales**

Movearse en una dirección.

- Basados en orientación.
- Vagar.

- **Apetitivos orientados por objetivo**

Movearse hacia un atractor.

- Objeto discreto.
- Superficie.

- **Aversivo/Proteccionista**

Evitar colisiones.

- Evitar objetos estacionarios.
- Eludir objetos móviles (esquivar, escapar).
- Atacar.

Categorización ^{2/4}

- **Seguir caminos**

Moveirse siguiendo un camino determinado.

- Seguimiento de rutas.
- Navegación en corredores.
- Seguimiento de líneas.

- **Posturas**

- Balancearse.
- Estabilidad.

- **Social/Cooperativo**

- Compartir.
- Buscar alimentos.
- Moveirse en manadas.

Categorización ^{3/4}

- **Teleoperación**
Coordinar con un operador humano.
 - Influenciar.
 - Cambios de comportamiento.
- **Perceptual**
 - Movimientos sacádicos.
 - Búsqueda visual.
 - Reflejos oculares.
- **Caminatas**
Para robots n-podos.
 - Control de pasos.

Categorización ^{4/4}

- Manipuladores
Control de brazos. Robots poli-articulados.
 - Alcanzar.
- Agarre/Destreza de manos
Manipulación de objetos.
 - Agarrar.
 - Envolver.

Expresión

Existen varios métodos para expresar comportamientos robóticos.

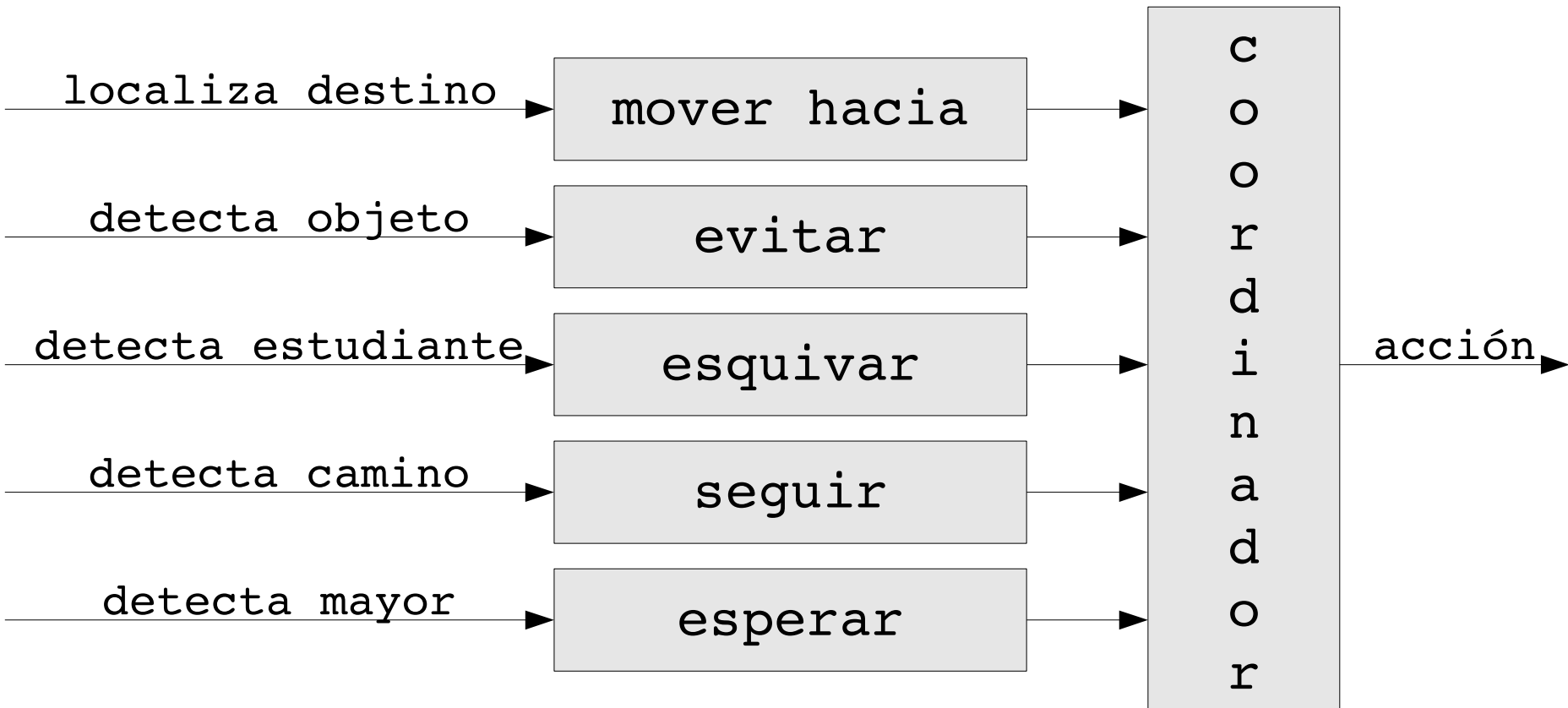
- Diagrama de estímulo-respuesta (*SR*)
- Notación funcional
- Diagrama de Receptores de Estado Finito (*FSA*).
- Métodos formales
Basados en modelos formales
 - Esquemas (*RS, Lyons & Arbid 1989*)
 - Autómatas situados (*Kaelbling & Rosenschein 1991*)

Diagrama SR ^{1/2}

- Todo comportamiento puede representarse como la respuesta que se obtiene a un estímulo concreto.
- Más intuitivo, menos formal.
- Se pueden visualizar gráficamente.



Diagrama *SR* _{2/2}



Notación funcional ^{1/2}

- Pueden aplicarse métodos matemáticos para describir las mismas relaciones utilizando notación funcional.
 - $b(s) = r$

Se interpreta: ante la presencia de un estímulo s el comportamiento b entrega la respuesta r .
- Menos intuitivo, más formal.
- En un sistema reactivo puro, b es independiente del tiempo debido a que la respuesta es instantánea e independiente de la historia.
- Suele permitir una rápida codificación en lenguajes funcionales.

Notación funcional ^{2/2}

```
coordinar-comportamientos(  
    mover_hacia(localiza_destino),  
    evitar(detecta_objeto),  
    esquivar(detecta_estudiante),  
    seguir(detecta_camino),  
    esperar(detecta_mayor)) =  
    respuesta_motora
```

```
coordinar-comportamientos(  
    coordinar-comportamientos(conj_comp_1),  
    coordinar-comportamientos(conj_comp_2),  
    coordinar-comportamientos(conj_comp_3)) =  
    respuesta_motora
```

Diagrama *FSA* ^{1/4}

- Útiles para representar agregaciones o secuencias temporales de comportamientos.
- Permiten explicitar la activación de un comportamiento en un momento dado y la transición entre ellos.
- Son poco útiles para describir un único comportamiento.
- Ha sido ampliamente utilizado para diseñar/expresar sistemas de control robótico.

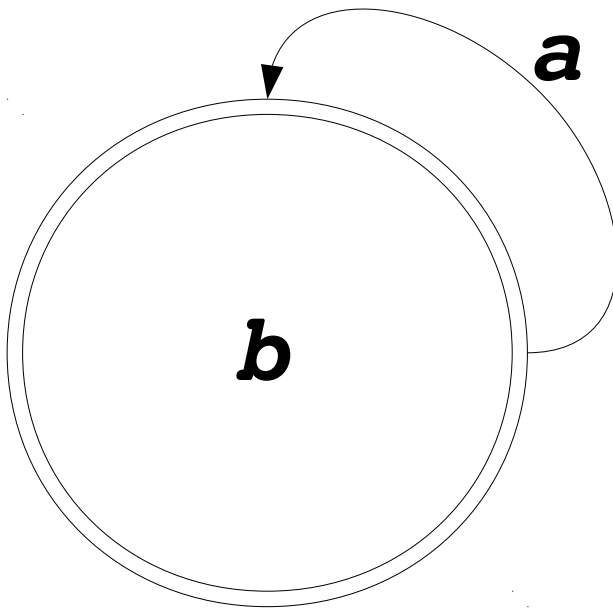
Diagrama FSA 2/4

- Un receptor de estado finito M puede representarse mediante una tupla:

$$(Q, \mu, q_0, F)$$

- donde Q representa el conjunto de estados, μ es la función de transición, q_0 representa el estado inicial y F el conjunto de estados destino.

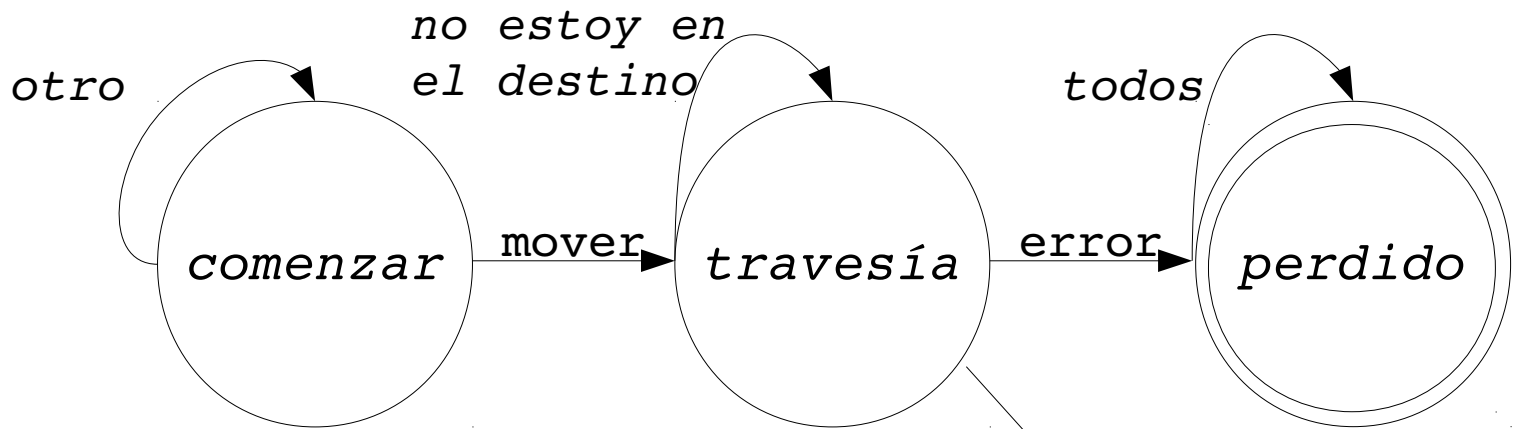
Diagrama FSA ^{3/4}



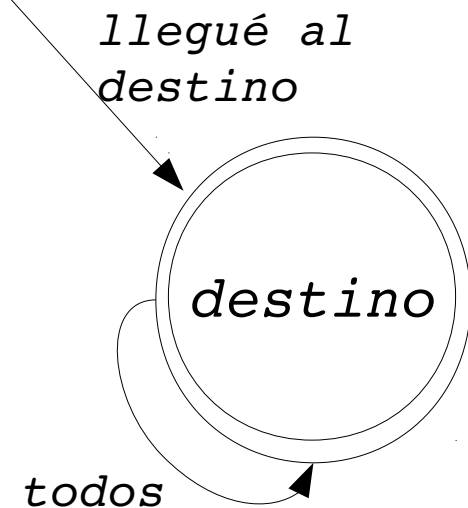
μ	Q	$input$	$\mu(q, input)$
	b	a	b

$$M = \{ \{b\}, \mu, b, \{b\} \}$$

Diagrama FSA 4/4



μ	Q	$input$	$\mu(q, input)$
	<i>comenzar</i>	<i>mover</i>	<i>travesía</i>
	<i>comenzar</i>	<i>otro</i>	<i>comenzar</i>
	<i>travesía</i>	<i>error</i>	<i>perdido</i>
	<i>travesía</i>	<i>n_e_e_d</i>	<i>travesía</i>
	<i>travesía</i>	<i>ll_d</i>	<i>destino</i>
	<i>destino</i>	<i>todos</i>	<i>destino</i>
	<i>perdido</i>	<i>todos</i>	<i>perdido</i>



Codificación del comportamiento ^{1/4}

- Crear una correspondencia funcional entre estímulos y respuestas motoras.
- Es necesario comprender la dimensionalidad de la respuesta motora del robot.
- La respuesta motora puede factorizarse en:
 - intensidad o fuerza de la respuesta.
Denota la magnitud. Suele relacionarse con intenciones o habituación.
 - orientación de la respuesta.
Denota la dirección de la acción. Requiere conocimiento sobre la cinemática del robot.
 - Ambas, pueden o no ser dependientes de la fuerza del estímulo.

Codificación del comportamiento 2/4

- Un comportamiento puede expresarse por medio de la terna: (S, R, β) .
 - $\beta = S \rightarrow R$
- R denota el recorrido de las posibles respuestas.
 - Asociado con:
 - Dimensión del espacio de configuraciones.
 - Grados de libertad del robot.
 - Restricciones cinemáticas afectan el movimiento del robot y pueden disminuir el espacio de configuraciones (Holonómicas, No-Holonómicas)

Codificación del comportamiento ^{3/4}

- S denota el dominio de todos los estímulos interpretables.
 - Su sola presencia es necesaria pero no suficiente para que exista respuesta motora
 - Cada estímulo es representado por el par (p, λ)
 - p : categoría
 - λ : fuerza
 - Discreta: binaria, difusa
 - Continua: real
 - ζ umbral por encima del cual los estímulos s de la categoría p generan una respuesta motora.
 - Perceptual triggers

Codificación del comportamiento 4/4

- β denota la correspondencia entre estímulos y respuestas ($\beta: S \rightarrow R$).
 - Null
 - Discreto
 - Binario
 - Basado en reglas: si-entonces, sistemas difusos
 - Continuo
 - Espacio de respuestas “infinito”
 - Artificial Potential Fields

Composición

- Comportamiento emergente.
 - Aparición de nuevas propiedades observadas a nivel del sistema. (Moravec, 1988)
 - Funcionalidades globales emergen a partir de la interacción paralela de comportamientos locales. (Steels, 1990)
 - Inteligencia que emerge a partir de la interacción de las componentes del sistema. (Brooks, 1991)
 - Se alcanzan funcionalidades emergentes en virtud de la interacción entre componentes que no fueron diseñados para brindarla. (McFarland & Bosser, 1993)

Coordinación

- Métodos competitivos
 - Arbitraje: *Winner-take-all*
 - Jerarquías: prioridades
 - *Action selection*
 - Votaciones
- Métodos cooperativos
 - Fusión
 - Potential fields

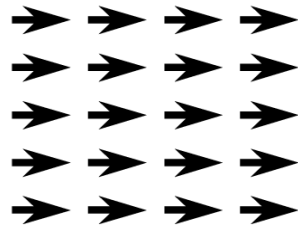
Campos de potencial (R. Arkin)

- Existen muchas implementaciones.
- Comportamientos como vectores.
- Comportamiento emergente a partir de la suma vectorial.
- La acción a tomar debe representarse como un campo de potencial.
- El campo de potencial puede verse como la fuerza que se realiza sobre el robot.

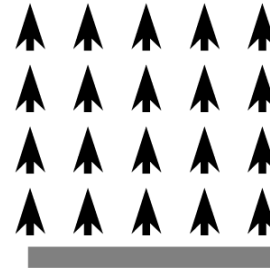
Campos de potencial

- El esquema motor de un comportamiento puede expresarse utilizando la metodología de campos de potencial.
- Un campo potencial puede ser primitivo o construido a partir de la suma de campos primitivos.
- El robot puede verse como una partícula.
- El robot siente la fuerza de cada comportamiento.
- Son continuos, cada punto del espacio representa un vector que el robot sentirá si está en ese punto.

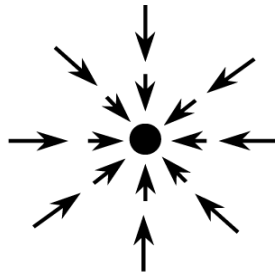
Cinco campos primitivos



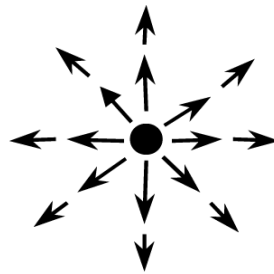
a



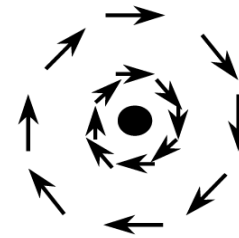
b



c



d

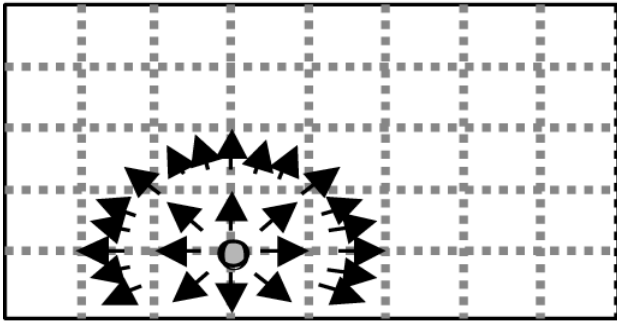


e

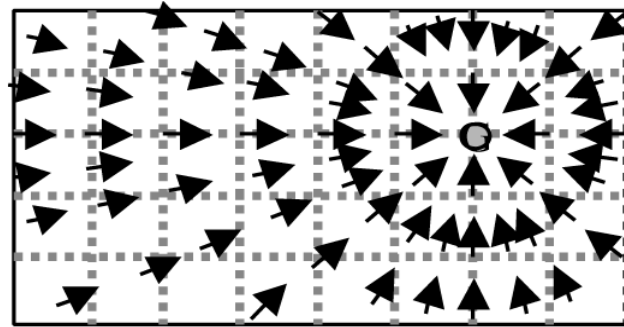
Campos y comportamientos

- Uniforme
 - Moverse en una dirección particular, seguir un corredor.
- Perpendicular
 - Evitar paredes
- Atracción
 - Moverse hacia el objetivo
- Repulsión
 - Evitar obstáculos
- Tangencial
 - Moverse a través de una puerta
- Aleatorio

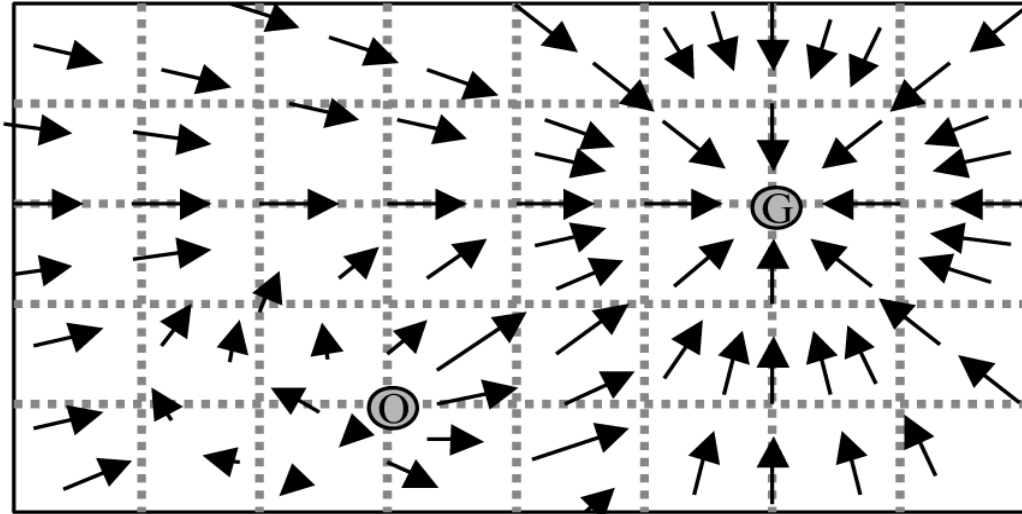
Ir a una posición (1/2)



a.

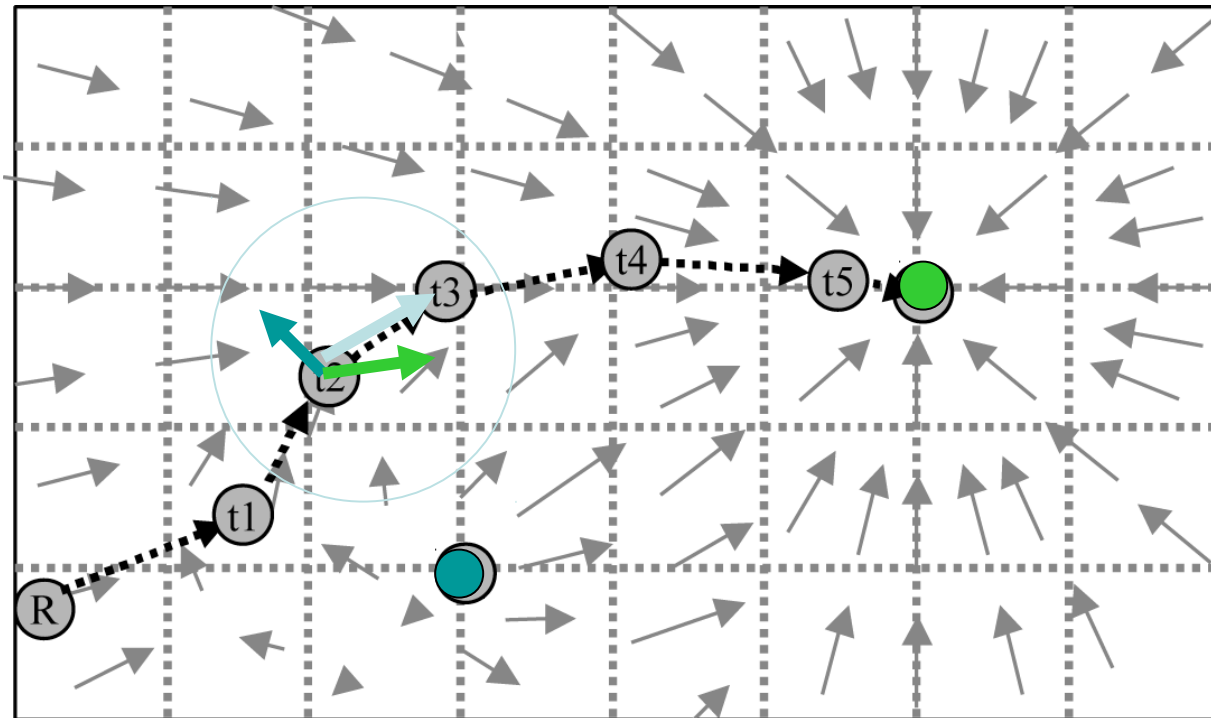


b.



c.

Ir a una posición (2/2)



Ventajas

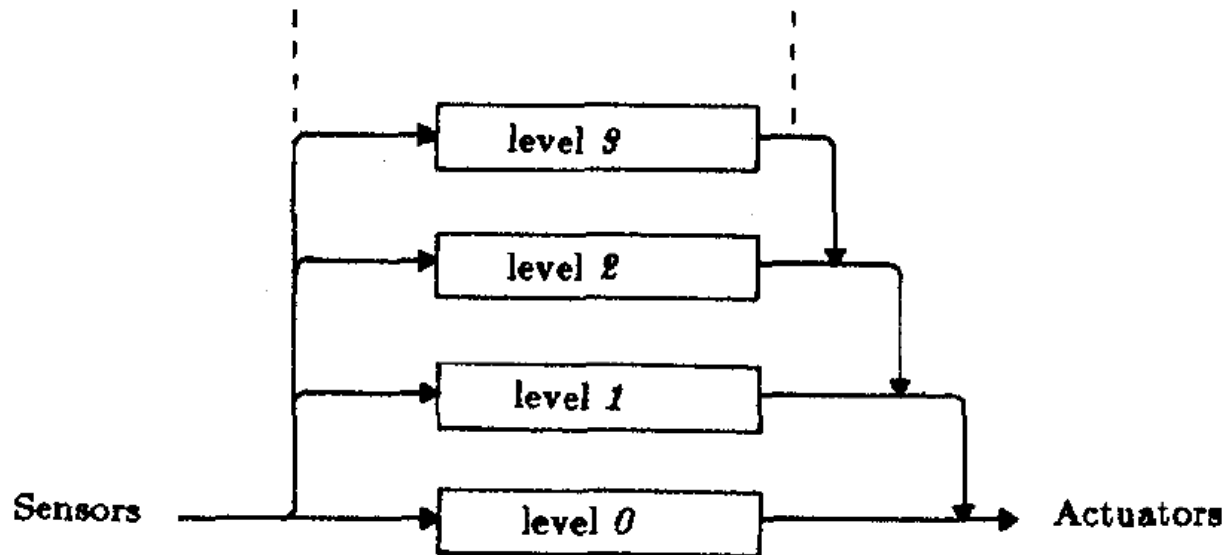
- Se puede visualizar gráficamente el comportamiento.
- Simple de implementar y combinar los comportamientos en lenguajes de programación.
- Mecanismo de combinación fijo.

Subsumption (R. Brooks)

- Un comportamiento es una red de módulos de sensado y actuación que realizan una tarea.
- Los módulos son implementados como máquinas de estado finito.
- Los módulos se agrupan en capas de competencia.
- Los módulos de una capa pueden sobrescribir o subsumir módulos de una capa inferior.
 - Supresión: sustituye la señal.
 - Inhibir: inhibe la transmisión de una señal.
- No utiliza ni mantiene ningún modelo del mundo.

Capas

- Metodología en capas.
 - Competencias
- Conecta sensores con actuadores en paralelo.

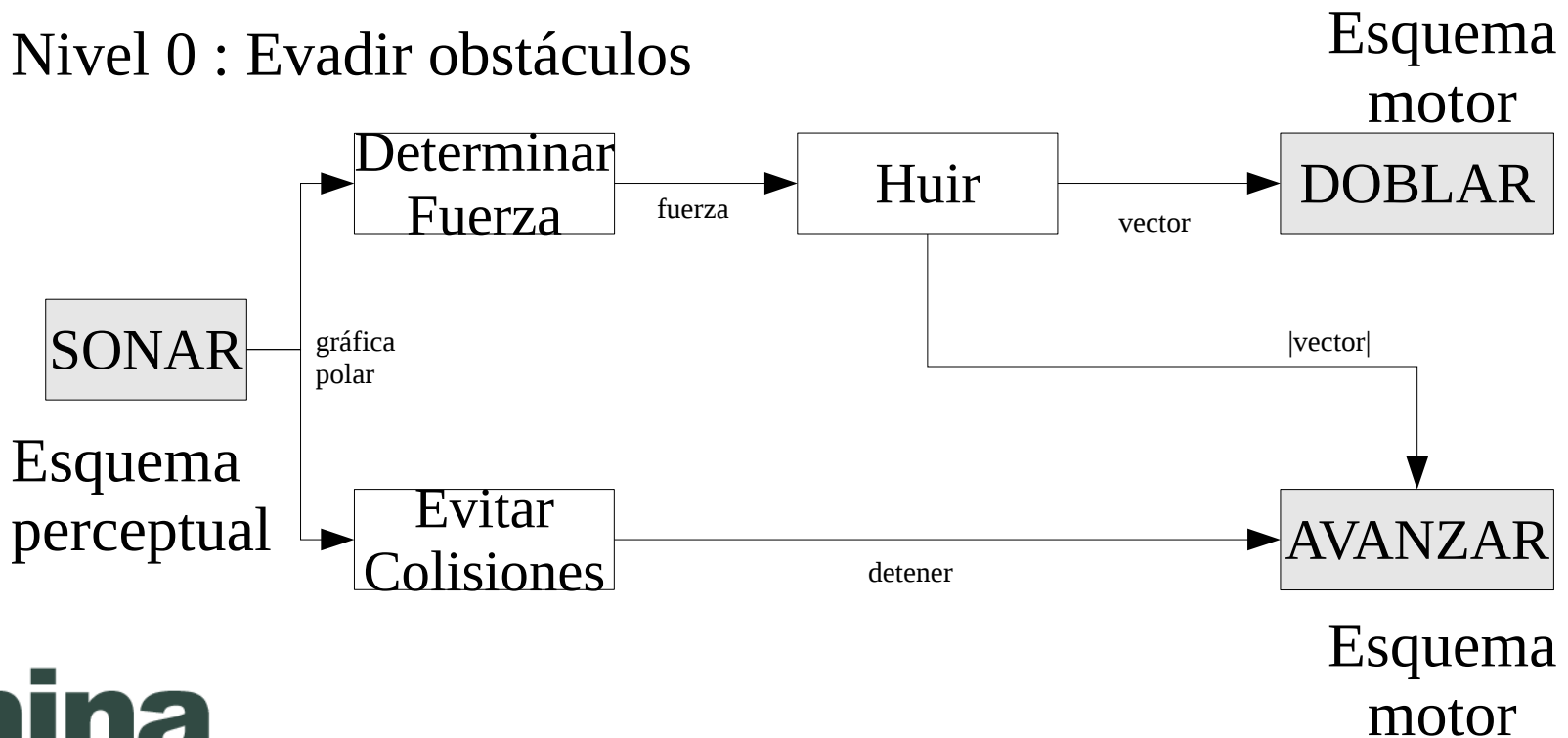


Interacción/Comunicación



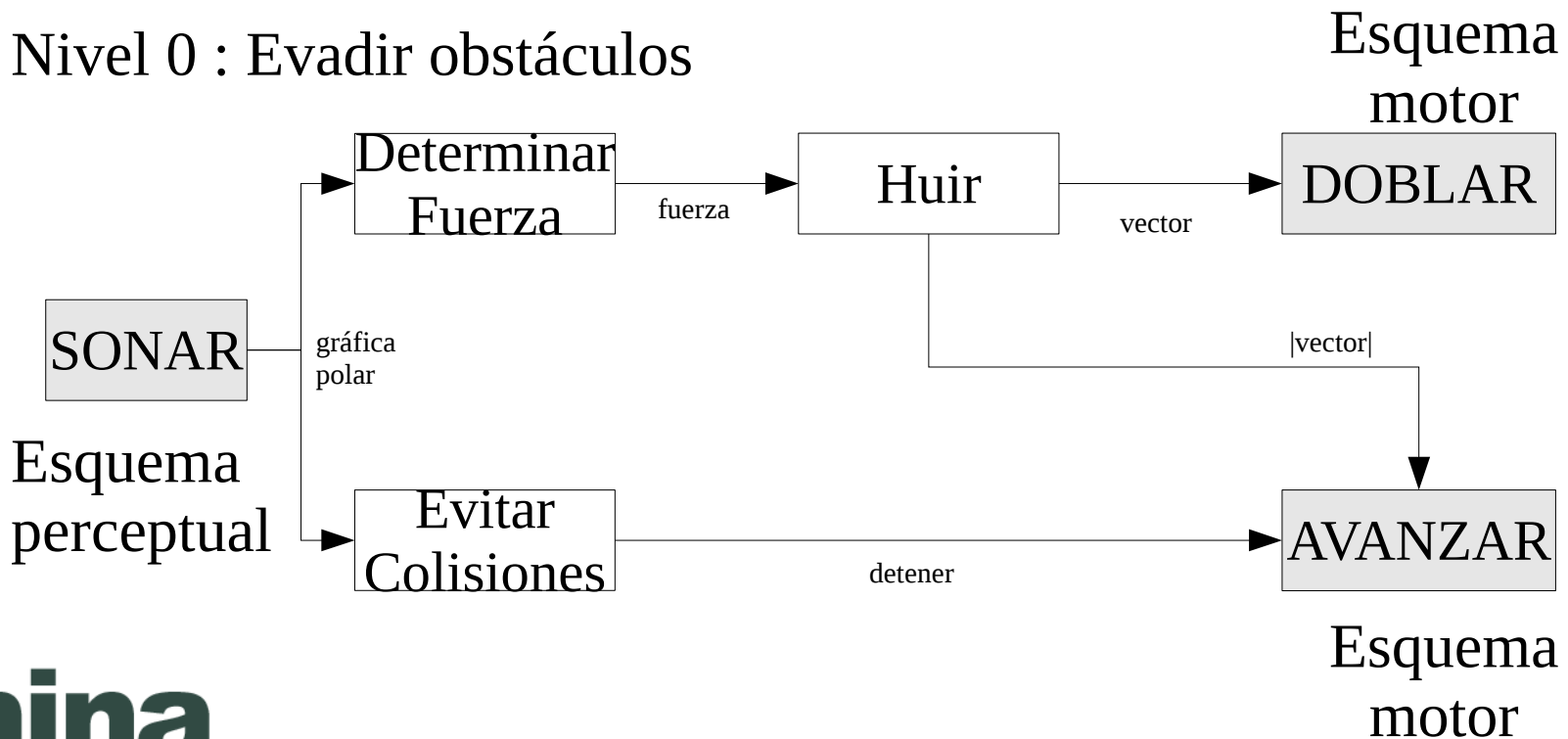
Ejemplo: Evadir obstáculos

Nivel 0 : Evadir obstáculos



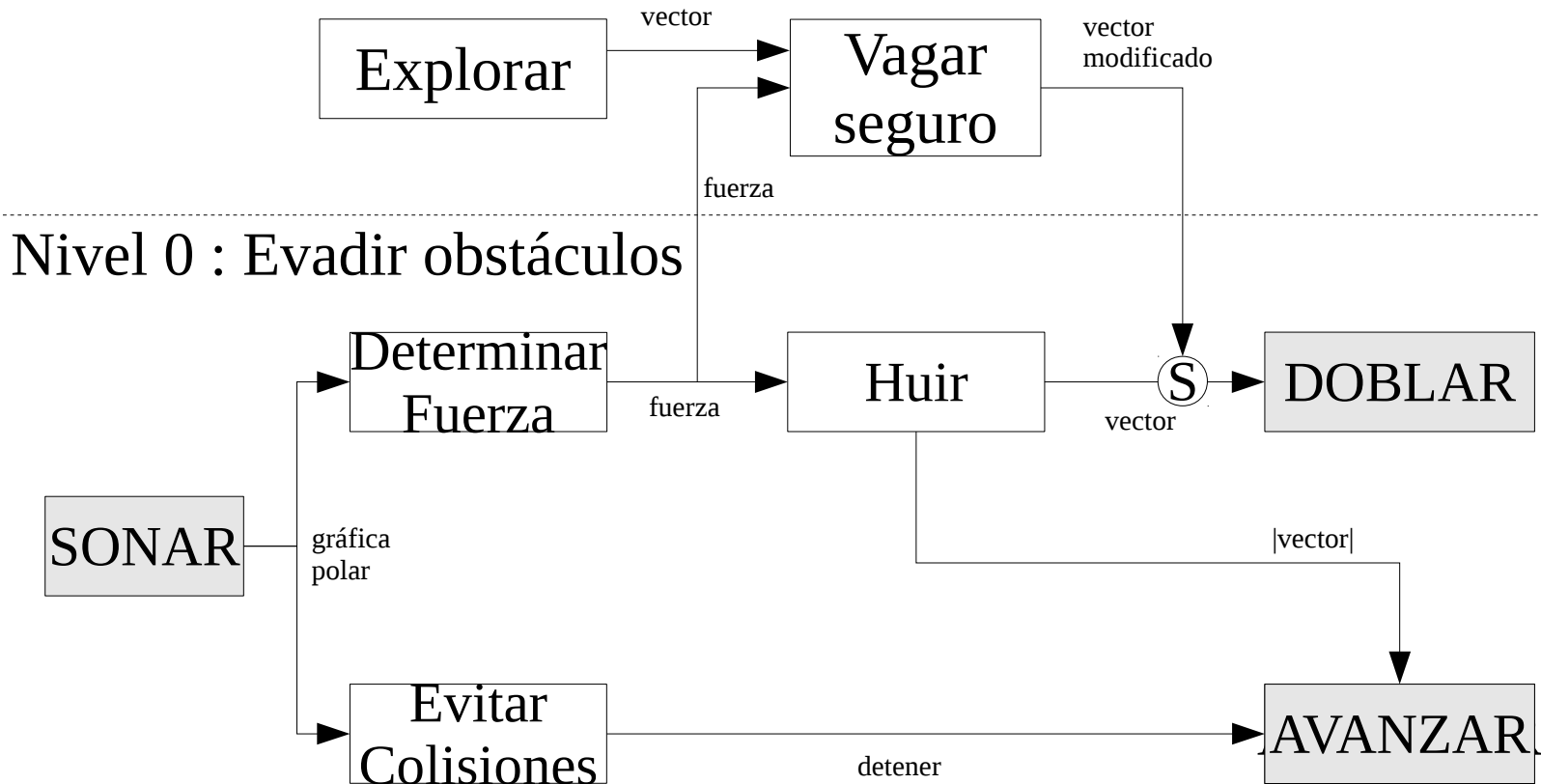
Ejemplo: Vagar

Nivel 0 : Evadir obstáculos



Ejemplo: Vagar

Nivel 1 : Vagar



Lejos

- Programación basada en comportamientos
 - package
 - lejos.robotics.subsumption
 - Interfaces
 - Behavior
 - Action(), suppress(), takeControl()
 - Clases
 - Arbitrator
 - start()
Determinar cuál es el comportamiento que debe activarse.
Suprimir el comportamiento activo (menos prioritario).
Ejecutar (activar) el comportamiento más prioritario.

Referencias

- Arkin R., “Behavior-Based Robotics”, Cap 3, MIT Press, 0262011654, 1998.
- Brooks, R., "A robust layered control system for a mobile robot". Robotics and Automation, IEEE Journal, 1986.