

# *Inteligencia Artificial*

## Agentes Inteligentes

Adaptado de: **CS 188: Artificial Intelligence**

University of California, Berkeley

[slides adapted from Dan Klein, Pieter Abbeel]

## **2. Agentes Inteligentes**

**2.1** Definición y tipos de Agentes

**2.2** Entorno de los Agentes

**2.3** Arquitectura de Agentes

# Agentes

- El objetivo final de la IA se podría decir que es el de obtener sistemas **racionales** que actúen del mismo modo a como lo haría la persona a la cual están sustituyendo.
  - Para ello, como no había una metodología definida, los investigadores **usaban criterios diferentes**.
  - El uso de diferentes criterios, **suponía un inconveniente** para el desarrollo y consolidación de esta nueva tecnología.
- Por ello, grandes investigadores como **Julián Vicente** y **Vincent Botti** trataron de definir una metodología universal: Agentes Inteligentes o Paradigma de Agentes (1999)



# Agentes: ¿qué son?

- Se basa en la **separación/desagregación** del problema en unidades autónomas capaces de **cooperar y coordinarse** en pro del logro de los objetivos del sistema en su conjunto.
- Fusión de tres corrientes, especialmente:
  - **Ingeniería del Software**: similitud con la ingeniería de software orientada a objetos, la cual desagrega la complejidad del problema en **entidades software independientes que son los objetos**.
    - Encapsulamiento, independencia
    - Mensajes entre objetos (comunicación)
    - Clases, herencia
  - **Sistemas Distribuidos**: es un sistema en el que los **componentes hardware o software** se encuentran en computadores **unidos mediante una red** y se **comunican** únicamente mediante paso de mensajes
    - Distribución de datos y procesos
    - Conectividad, Redes, Protocolos
    - Interoperabilidad
    - Internet
  - **Inteligencia Artificial**: Conocimiento (representación del entorno), razonamiento, aprendizaje, visión ....

# Objeto VS Agente

## ➤ Objetos:

- Los objetos tienen capacidad para el paso de mensajes a otros objetos
  - Ejemplo: **Java**
    - Métodos **privados y públicos**
    - El objeto conocen su estado, pero **no tiene control total sobre su comportamiento**
    - Un objeto **no puede impedir que otros objetos utilicen sus métodos públicos**

## ➤ Agentes:

- **Se comunican** con otros agentes y les **solicitan que ejecuten acciones por ellos**
- Los objetos siempre **hacen lo que se les pide**, los agentes **no**.
- En O.O. no hay similitud/analogía a ser **reactivo, pro-activo** o **social**
- En caso de **multiagentes**, cada agente tiene su hilo de ejecución.

# Agentes: definición

## ➤ **Rusell & Norving:**

- Un agente es cualquier cosa que pueda ver su entorno a través de sensores y actuar en su entorno a través de efectores.

## ➤ **H. S. Nwana:**

- Componente de software y/o hardware el cual es capaz de actuar de manera precisa con el fin de realizar tareas en representación de su usuario.

## ➤ **Maes:**

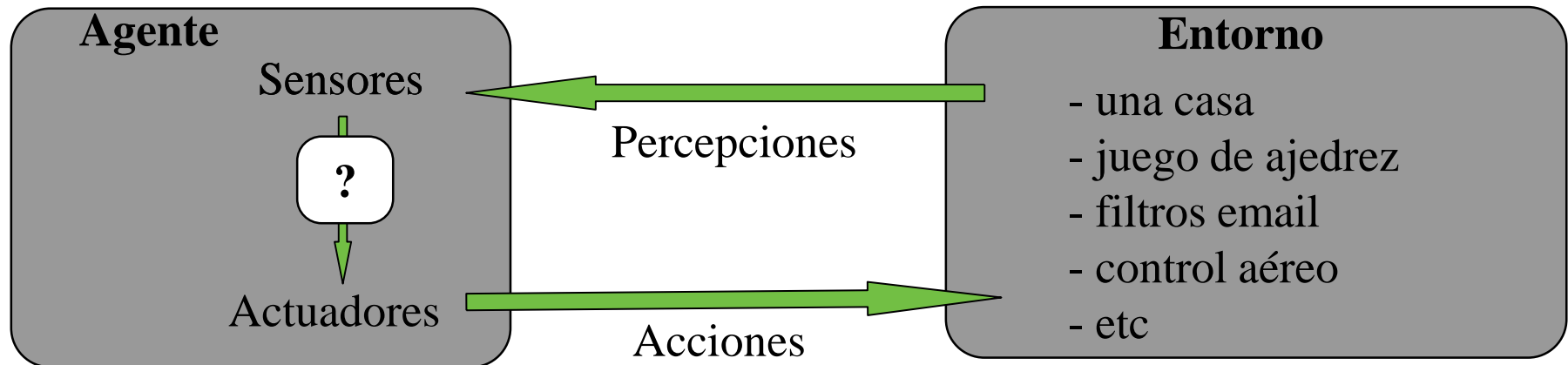
- Un agente autónomo es un sistema computacional que habita en un entorno dinámico complejo, percibiendo y actuando autónomamente en este entorno, y haciendo esto para realizar un conjunto de objetivos o tareas para los cuales fueron diseñados.

## ➤ **Wooldridge & Jennings:**

- Un agente es un sistema computacional que está situado en algún ambiente, y que es capaz de actuar autónomamente en dicho ambiente con el fin de cumplir sus objetivos.

# Agentes y entorno

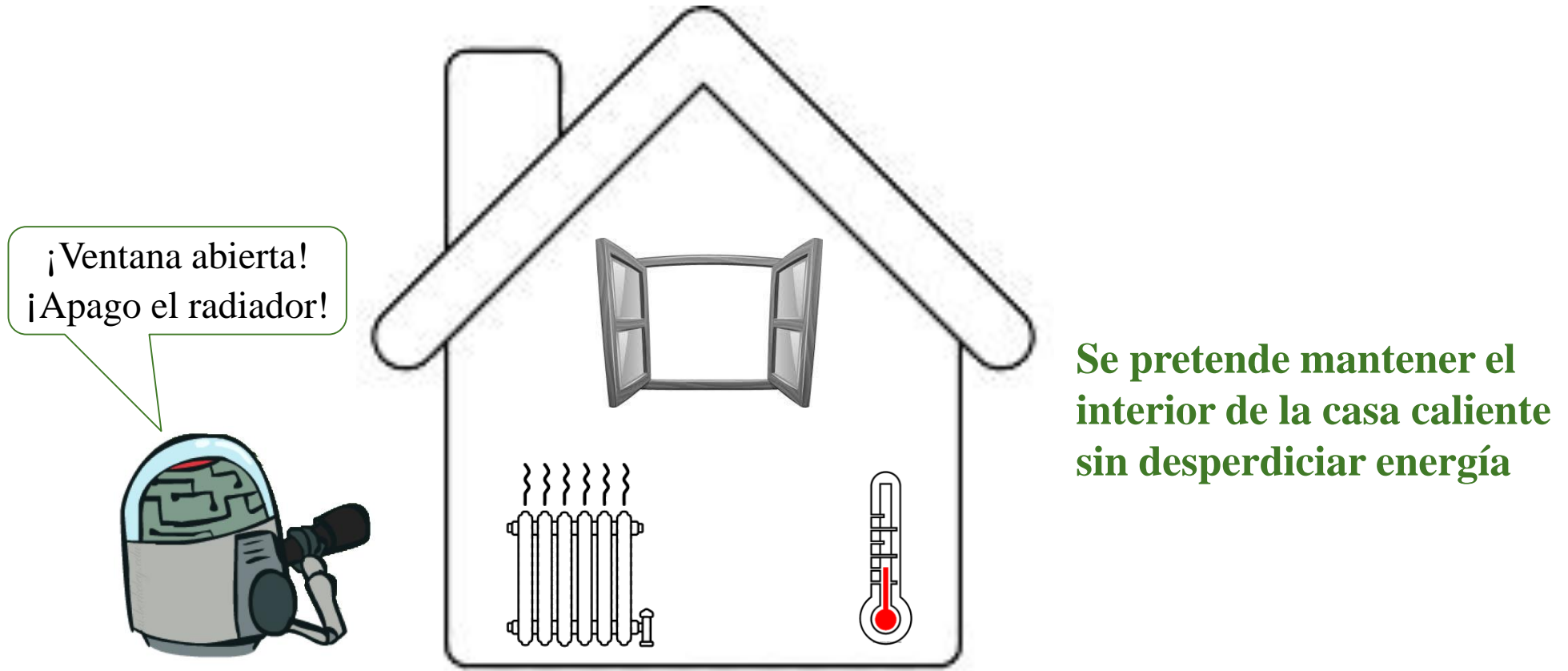
- Un agente **percibe su entorno** a través de sensores y **actúa sobre él** a través de **actuadores**



- El objetivo de los agentes es ejecutar tareas complejas en beneficio de los usuarios, que de otra manera serían difíciles de lograr
- Dentro de un agente se encuentran programas de software, pero no todos los programas informáticos son agentes. ¿Diferencia? Las principales características de un agente (Jennings & Wooldridge):
  - **Autonomía:** un agente puede ser capaz de tener el control sobre sus acciones y sobre su estado interno (el agente puede tener conocimiento).
  - **Flexibilidad:** reactivo, pro-activo y comunicativo

# Reactividad

- Los agentes perciben su entorno (mundo real, un usuario, una colección de agentes, una base de datos, Internet, etc.) y responden en la forma adecuada a los cambios que se han producido en él, sin precisar de la intervención de un tercero.



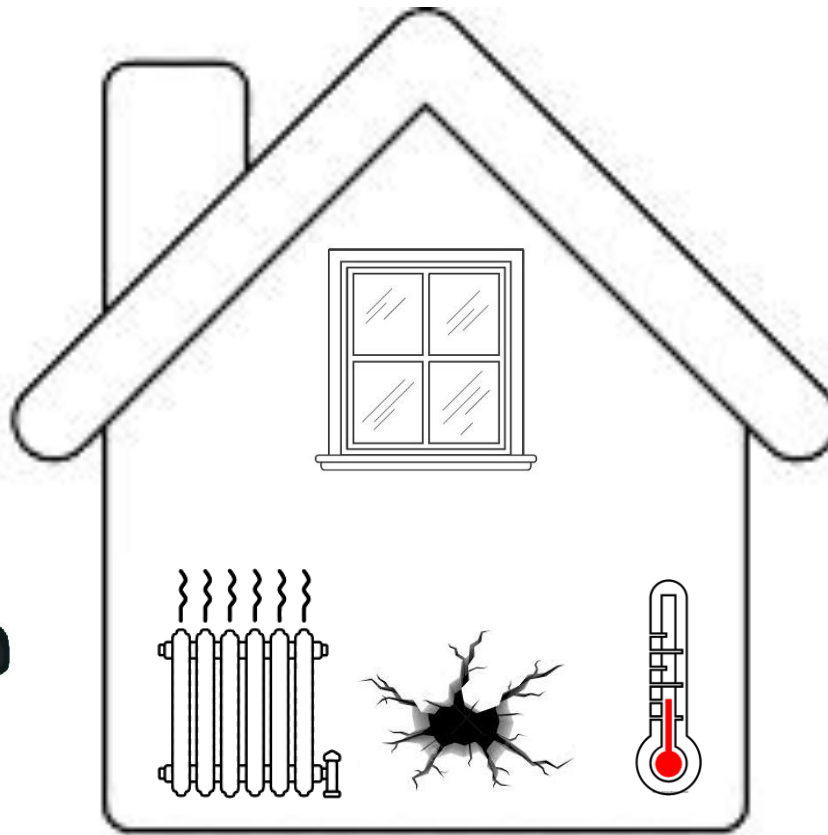
- No poseen modelos simbólicos de su entorno; en su lugar **actúan y responden a los estímulos** que presenta el estado actual del entorno en el que están embebidos.



# Pro-actividad

- Los agentes tiene la **capacidad de anticiparse al desenlace de los hechos, tomando iniciativas**, siempre orientando su comportamiento a la consecución de los objetivos que tiene asignados. De este modo, los agentes no sólo actúan en respuesta a los estímulos del entorno.

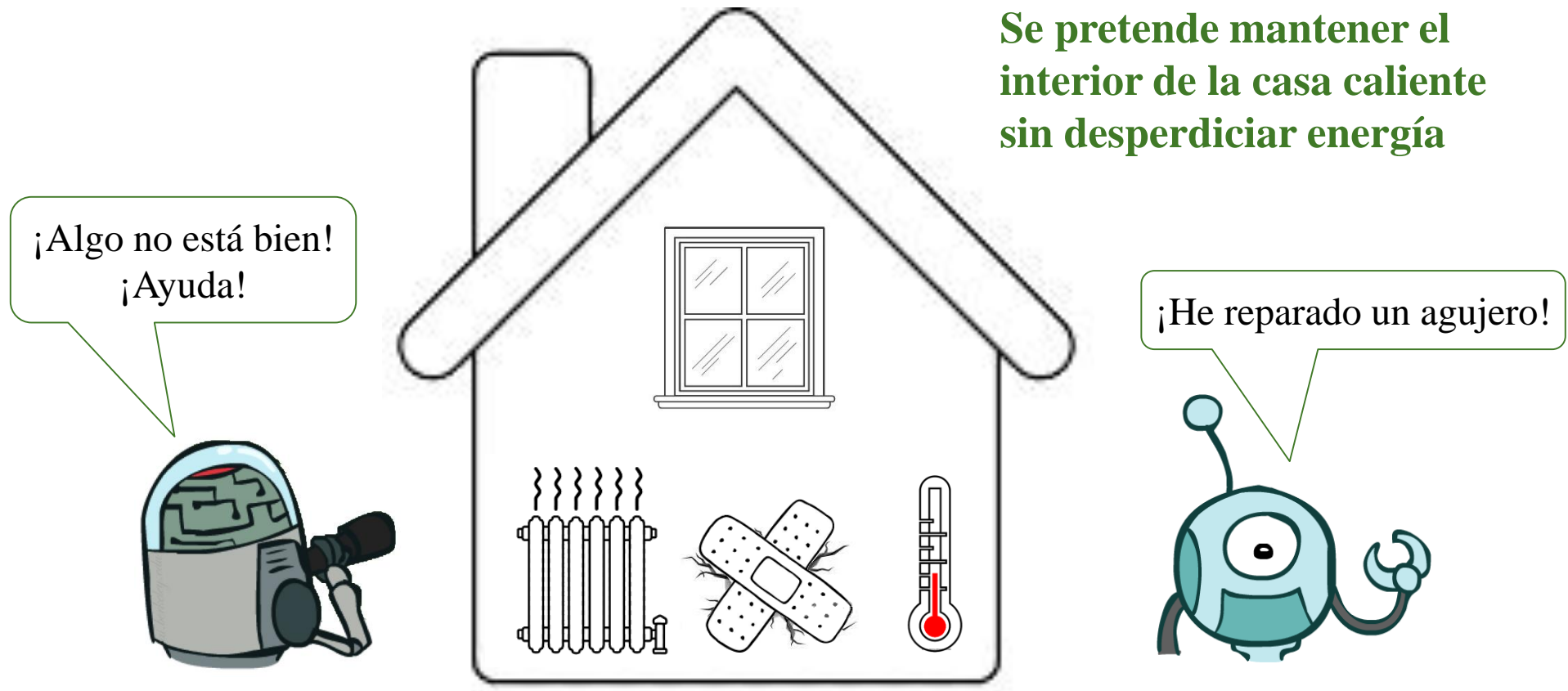
¿Qué sucede?  
¡Ventana cerrada pero  
baja la temperatura!  
¡Subiré el radiador!



**Se pretende mantener el interior de la casa caliente sin desperdiciar energía**

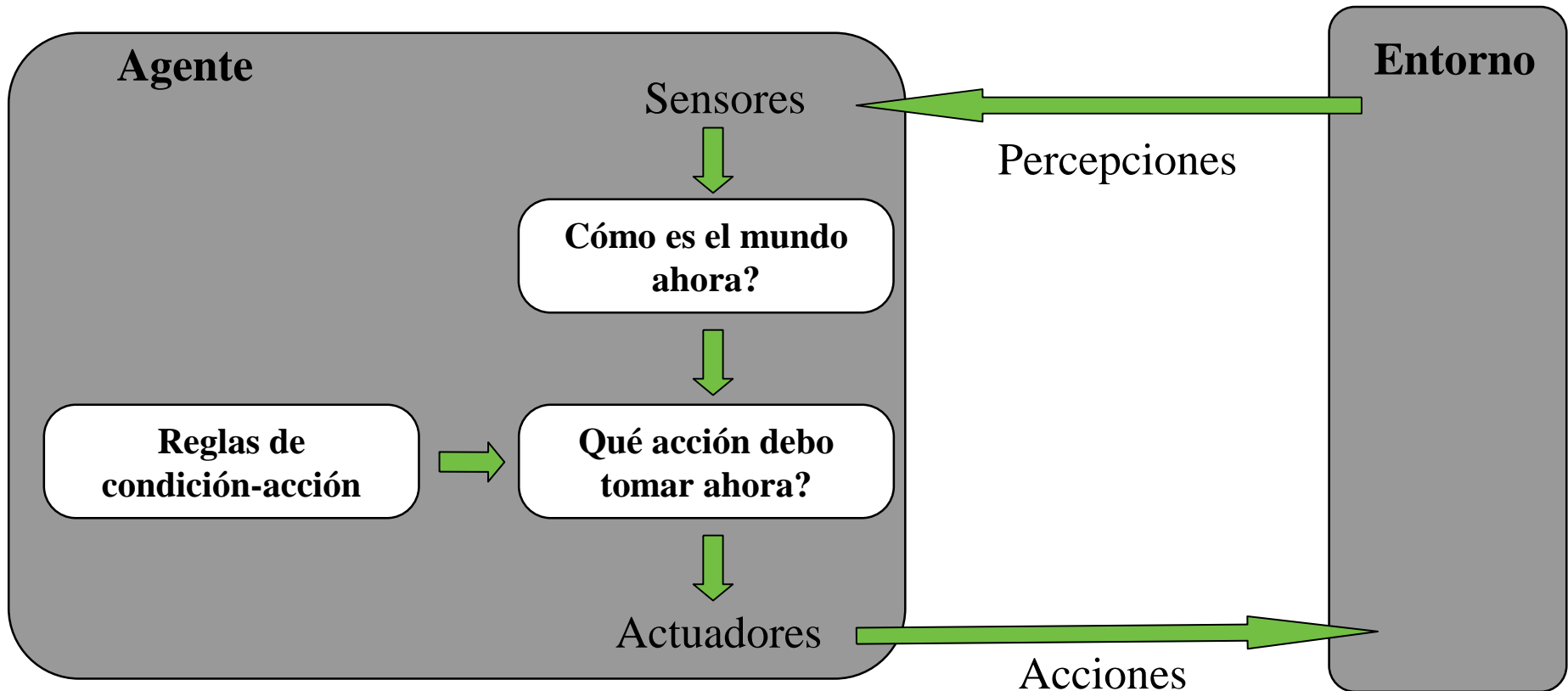
# Comunicación

- Los agentes **pueden interactuar** (cooperar, coordinar, negociar) cuando sea necesario **con otros agentes** con el fin de **completar su propia tarea y ayudar a completar sus tareas a otros agentes**. Un agente debe ser capaz de poder entablar comunicación con otros agentes a través de un lenguaje de comunicación de agentes **ACL** (Agent Communication Language)



# Estructura de un agente Reactivo Simple

- Este agente **basa sus acciones dependiendo** de las **percepciones actuales** sin importar las **percepciones anteriores**, poseen inteligencia muy limitada.



- Las reglas **condición-acción** permiten establecer la **conexión entre percepción y acción**.
- Se indica el **estado interno, único en un momento dado** del proceso de decisión.
- Se indica la base de conocimiento en forma de reglas de producción

# Diseño de un agente Reactivo Simple

```
function Simple-Reflex-Agent(percept) returns action
static: rules, a set of condition-action rules
    state ← Interpret-Input(percept)
    rule ← Rule-Match(state, rules)
    action ← Rule-Action[rule]
return action
```

- **Interpret-input:** genera una descripción abstracta del estado mostrado por la percepción.
- **Rule-match:** proporciona una regla del conjunto que satisface la percepción
- **Problemas:** bucles infinitos en entornos parcialmente observables (**no tienen memoria**)
- **Solución:** escoger acciones aleatoriamente ➡ **no es racional**

# Ejemplo de un agente Reactivo Simple

- **Robot móvil que se mueve en un entorno evitando obstáculos**
  - Tres sensores de proximidad: **frontal**, **derecho** e **izquierdo**.
  - Tres posibles acciones: **avanzar**, **girar derecha** y **girar izquierda**.
- El comportamiento se puede representar mediante una tabla
  - El módulo de comportamiento es tan sencillo como una búsqueda en una tabla.

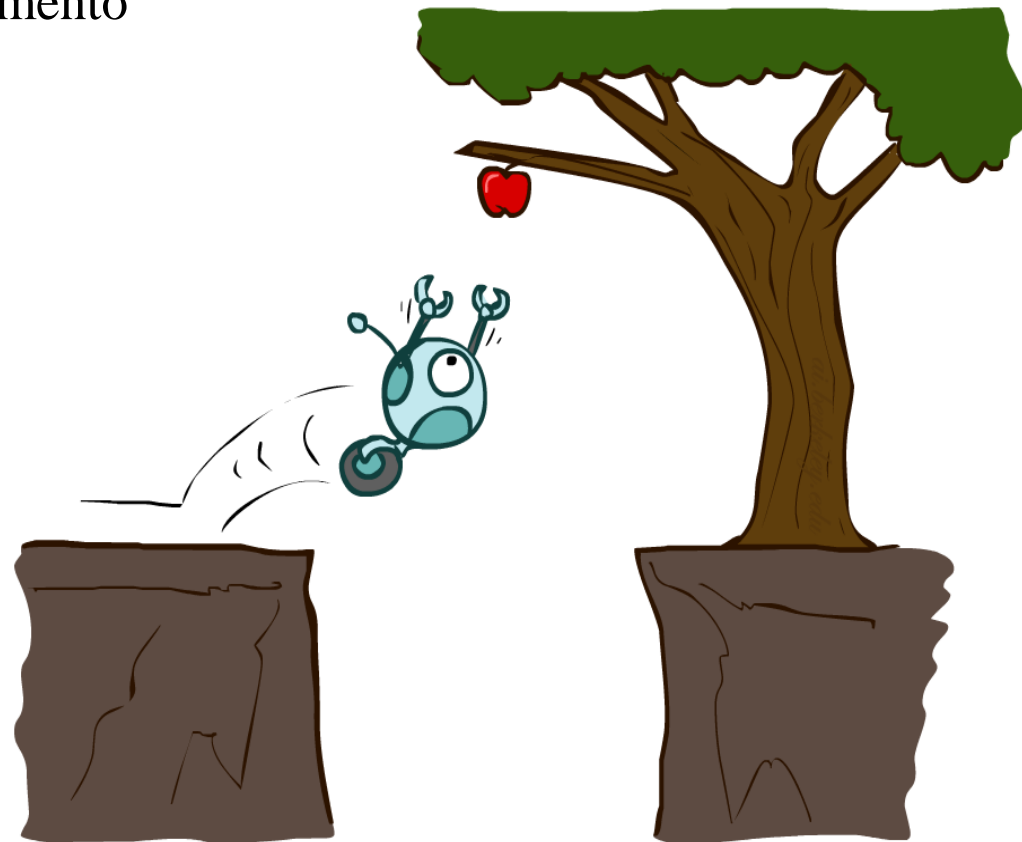
| Sensor frontal | Sensor derecho | Sensor izquierdo | Acción          |
|----------------|----------------|------------------|-----------------|
| libre          | libre          | libre            | avanzar         |
| no libre       | libre          | libre            | girar derecha   |
| no libre       | no libre       | libre            | girar izquierda |
| ...            | ...            | ...              | ...             |

# Agentes Reactivos o Reflex

## ➤ Agentes Reactivos / Reflex:

- Eligen la acción basándose en la percepción actual
- **No consideran las consecuencias futuras de sus acciones**
- Consideran cómo **ES** el mundo en ese momento

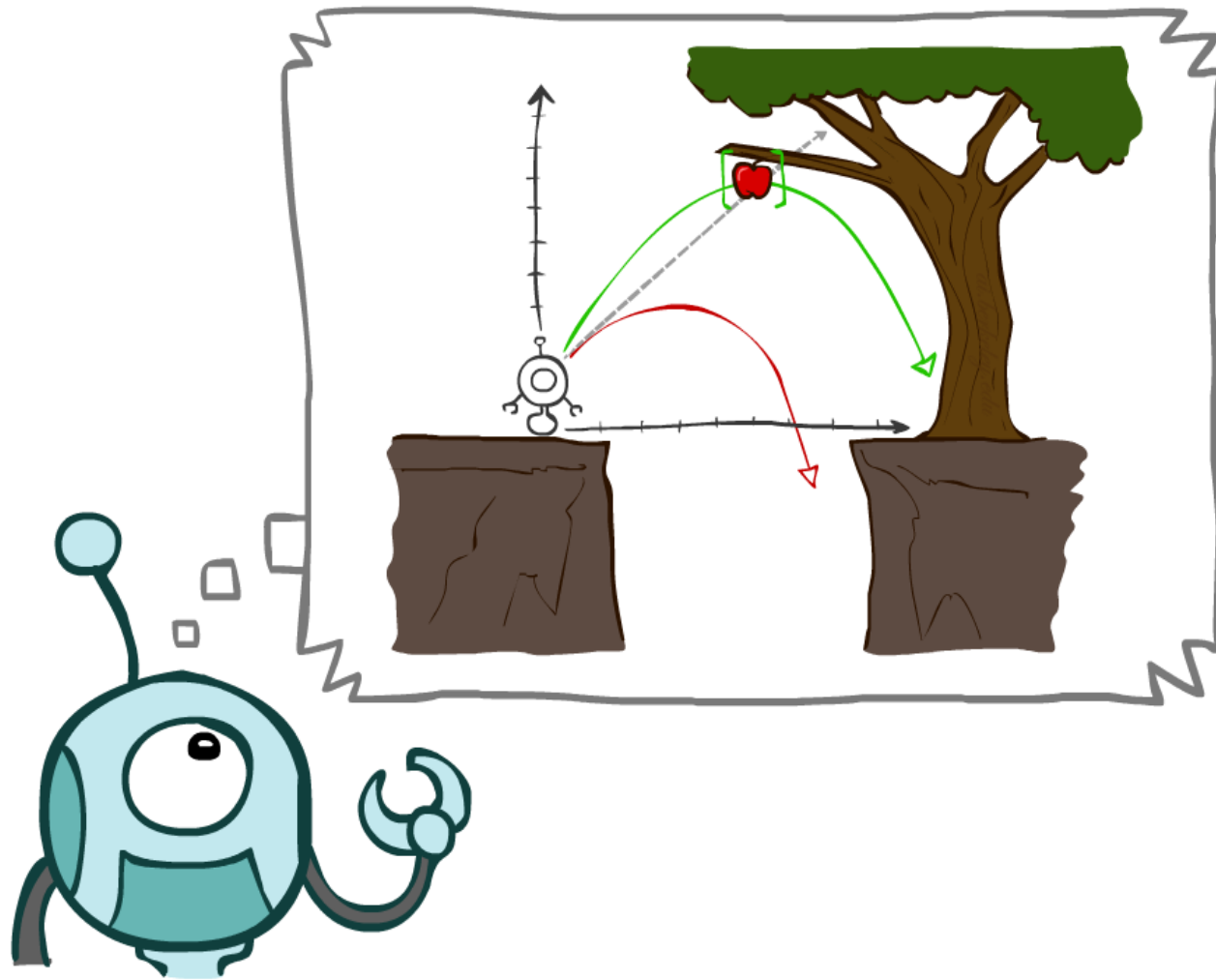
## ➤ ¿Puede un agente reflex ser **racional**?



# Racionalidad

- Un agente racional elige acciones que **maximizan la utilidad esperada**
- **Inicialmente:** agentes que tienen un **objetivo**, y un **coste**
  - Por ejemplo, alcanzar un objetivo con el coste mínimo
- **Posteriormente:** agentes que tienen **utilidades** numéricas, **recompensas**, ...
  - Por ejemplo, tomar acciones que maximizan la recompensa total respecto al tiempo (ejemplo: máximo beneficio en €)

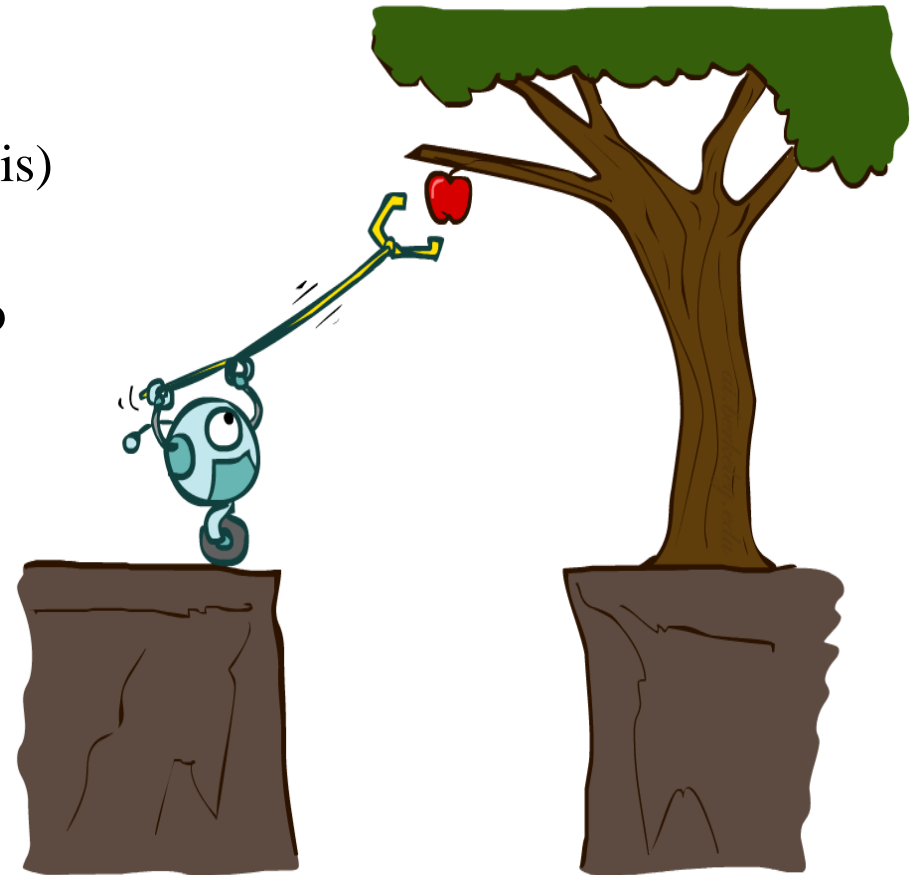
# Agentes que planifican





# Agentes que planifican

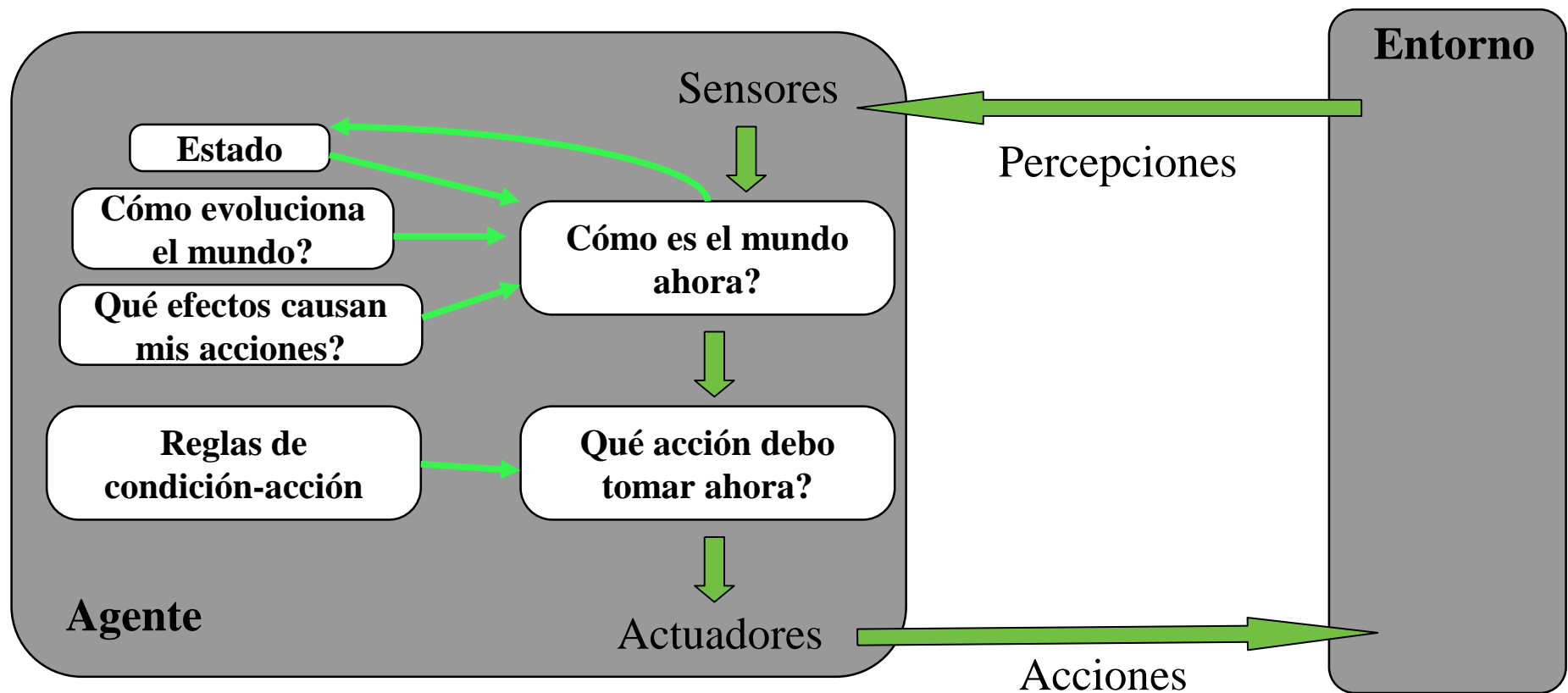
- Agentes que planifican:
  - Preguntan “qué pasa si ...”
  - **Decisiones basadas en** (planteando hipótesis) **consecuencias de acciones**
  - **Deben tener un modelo** de cómo el mundo evoluciona en respuesta a acciones
  - Considera cómo **SERÍA** el mundo
- Planificación **óptima** vs. **completa**
- Planificación vs. replanificación



# Tipos de Agentes

- Agentes **con memoria (modelos)**
  - con un estado interno que sirve para seguir la pista a los estados pasados de su mundo.
- Agentes **focalizados en encontrar la meta**
  - además de disponer de información sobre el estado, tienen información sobre situaciones deseables para alcanzar la meta. Lógicamente, los agentes de este tipo tienen bajo consideración eventos del futuro.
- Agentes **focalizados en mejorar la utilidad de sus acciones**
  - basan su decisión en la teoría axiomática clásica de la utilidad para actuar racionalmente.

# Estructura de Agente reactivo basados en modelos



- La forma de manejar **la visibilidad parcial del entorno**, es **almacenar información** de las partes del mundo que no pueden ver ➡ **mantener un estado interno**.
- Esto, requiere **codificar dos conocimientos**:
  - La **evolución** del mundo
  - Saber **cómo afectan** las acciones

# Diseño de un agente basado en modelos

```
function Reflex-Agent-With-State(percept) returns action
static: rules, a set of condition-action rules, state,
           a description of the current world

    state ← Update-State(state, percept)
    rule ← Rule-Match(state, rules)
    action ← Rule-Action[rule]
    state ← Update-State(state, action)
return action
```

- El conocimiento del **estado interno no siempre es suficiente**:
  - Al elegir entre dos rutas alternativas (en una intersección: ¿girar o no girar el volante del coche?) no se alcanzará el objetivo final.
  - **Se requiere conocer la meta a lograr** para finalizar la tarea con éxito

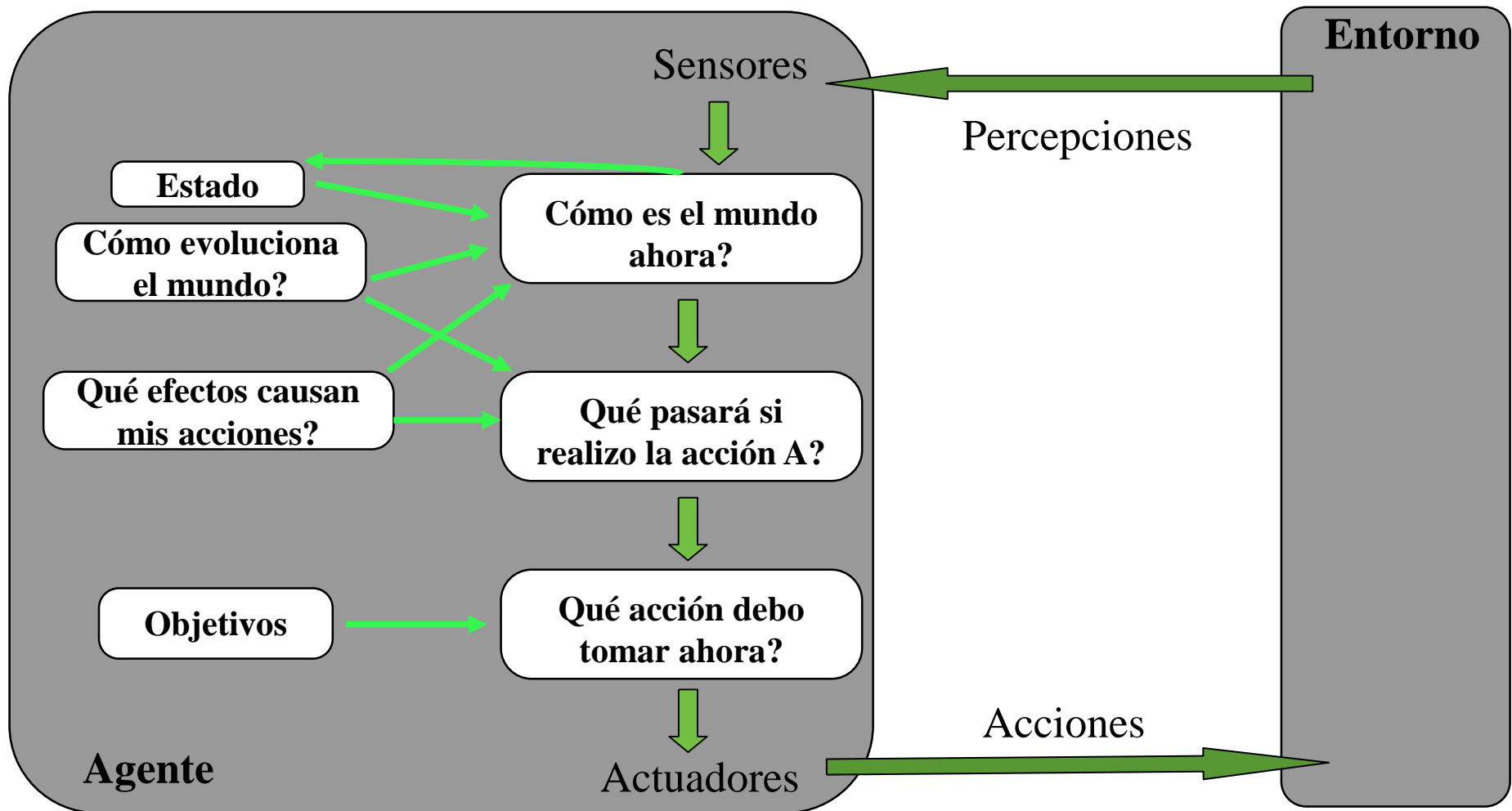
# Ejemplo de un agente basado en modelos

- **Robot móvil que se mueve en un entorno evitando obstáculos**
  - Tres sensores de proximidad: **frontal**, **derecho** e **izquierdo**.
  - Tres posibles acciones: **avanzar**, **girar derecha** y **girar izquierda**.
- El comportamiento se puede representar mediante una tabla
  - El módulo de comportamiento es tan sencillo como una búsqueda en una tabla.

| Sensor frontal | Sensor derecho | Sensor izquierdo | Acción          |
|----------------|----------------|------------------|-----------------|
| libre          | libre          | libre            | avanzar         |
| no libre       | libre          | libre            | girar derecha   |
| no libre       | no libre       | libre            | girar izquierda |
| ...            | ...            | ...              | ...             |

- Un mapa de los obstáculos y paredes de la habitación creado a partir de la historia de percepciones y acciones (movimientos).
- Al disponer de un mapa, las acciones del robot serían más lógicas (por ejemplo, podría buscar la dirección en la que los obstáculos se encuentren más alejados).

# Estructura de Agentes basados en objetivos



- El agente basado en objetivos y modelos, **almacena información del estado del mundo y del conjunto de objetivos**: es capaz de seleccionar la acción que eventualmente le guía a los objetivos. Pero, es suficiente esta estructura?

# Diseño de un agente basado en objetivos

```
Input percept
state ← Update-State(state, percept)
goal ← Formulate-Goal(state, perf-measure)
search-space ← Formulate-Problem (state, goal)
plan ← Search(search-space, goal)
while (plan not empty) do
    action ← Recommendation(plan, state)
    plan ← Remainder(plan, state)
    output action
end
```

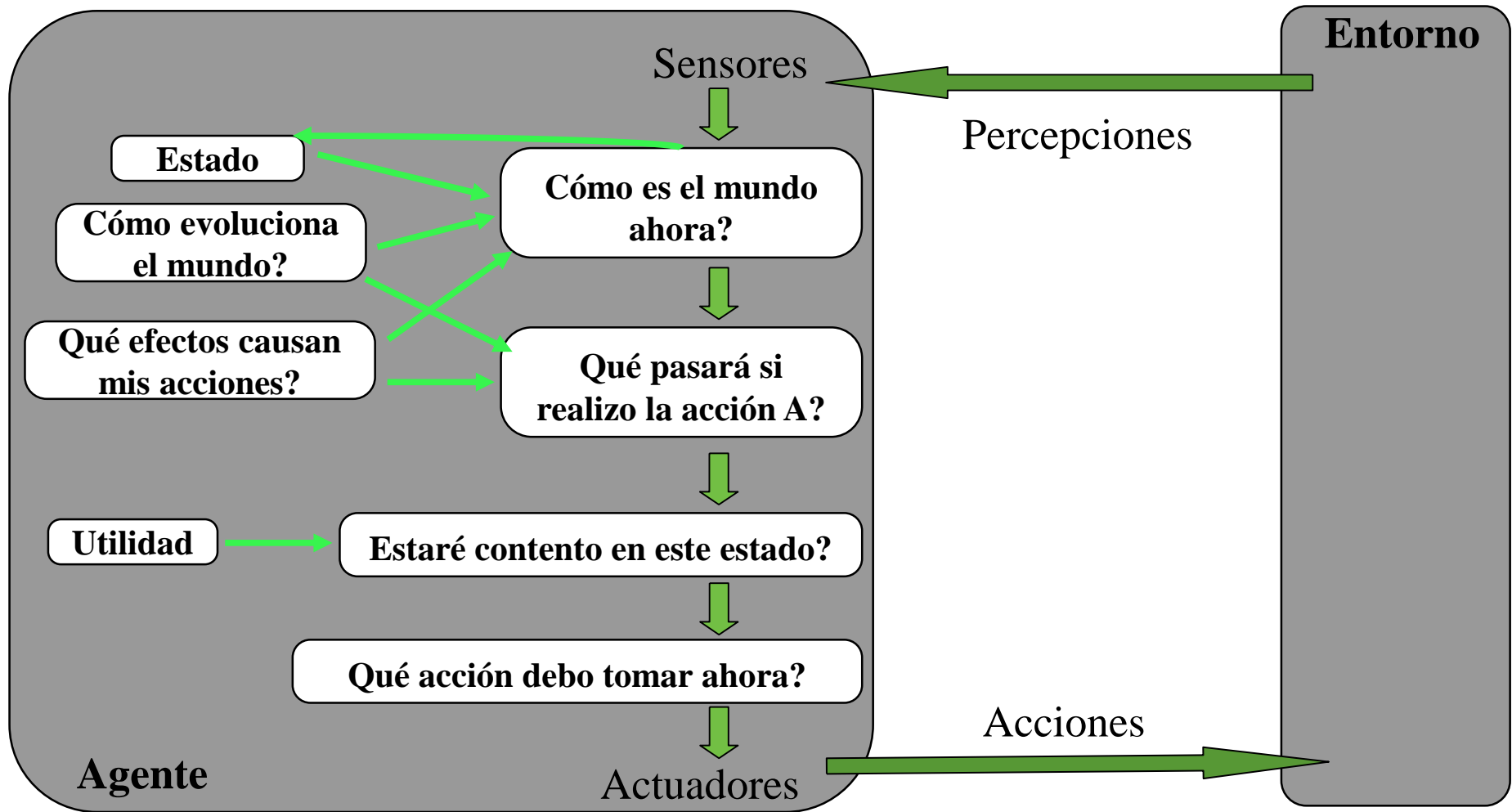
- Razonando acerca de acciones:
  - un agente reactivo solo actúa basado en conocimientos precomputados (reglas)
  - la búsqueda y la planificación ayudan a razonar acerca de qué acción logra la meta
  - el agente es menos eficiente pero más adaptativo y flexible
- Cómo escoger la mejor regla? Cómo argumentarlo?
  - hay que maximizar la utilidad de las acciones: elegir aquella que logra la mejor meta de todas.
- Modelado del efecto de las acciones: a cada par (estado, acción) se le asocia un efecto (un cambio en el estado). **Estado\_nuevo**=f(estado\_anterior, acción) **Meta**: estado final deseado para el sistema.

# Ejemplo de un agente basado en objetivos

- **Robot móvil que se mueve en un entorno evitando obstáculos**
  - Tres sensores de proximidad: **frontal, derecho e izquierdo**.
  - Tres posibles acciones: **avanzar, girar derecha y girar izquierda**.
- Las acciones **NO** se asocian al estado del entorno mediante una tabla.
- Un mapa de los obstáculos y paredes de la habitación creado a partir de la historia de percepciones y acciones (movimientos).
- Un **elemento** que **permite conocer el efecto de cada acción**.
- Una **meta o situación final a alcanzar** por el robot.
- El robot busca en cada estado una acción que permita llegar al punto de destino (o acercarse a él).



# Estructura de Agentes basados en utilidad



- El agente basado en **utilidad y modelos**, utiliza un **modelo** y una **función de utilidad** que calcula sus **preferencias entre estados** del mundo.
- Después **selecciona la acción** que le lleve a **alcanzar la mayor utilidad esperada** (media de estados posibles, ponderado con la probabilidad del resultado, etc)

# Agentes basados en utilidad

- Función utilidad:
  - es un mapeo de estados bajo la forma de números reales
  - lleva a decisiones racionales en dos tipos de situaciones:
    - evaluación de compromisos entre objetivos en conflicto
    - evaluación de metas en conflicto
- Los **objetivos no bastan** para generar una conducta de **alta calidad**.
  - Ejemplo: para llegar a Gasteiz puedo ir por ¡muchos caminos!
  - Todos me llevarán a Gasteiz, pero .... ¿**Hay caminos más cortos?**
  - **OJO!** Quizás el objetivo es terminar en Gasteiz: **no es necesaria la utilidad** = más complejo!
- Los objetivos permiten establecer una distinción entre estados “**felices**” e “**infelices**”.
- Si se prefiere un estado en lugar de otro, se dice que ese estado ofrece mayor “**utilidad**” al agente.
- **Utilidad:**
  - Función que caracteriza el grado de satisfacción
  - **PRINCIPIO DE MÁXIMA UTILIDAD ESPERADA**

# Ejemplo de un agente basado en utilidad

- **Robot móvil que se mueve en un entorno evitando obstáculos**
  - Tres sensores de proximidad: **frontal**, **derecho** e **izquierdo**.
  - Tres posibles acciones: **avanzar**, **girar derecha** y **girar izquierda**.
- Las acciones **NO** se asocian al estado del entorno mediante una tabla.
- Un mapa de los obstáculos y paredes de la habitación creado a partir de la historia de percepciones y acciones (movimientos).
- Un elemento que permite conocer el efecto de cada acción.
- Una meta o situación final a alcanzar por el robot.
- **Una valoración de la utilidad de cada estado.**
- El robot ejecuta en cada estado **la mejor acción** en función del acercamiento al punto de **destino** (muy similar al agente basado en objetivos).

# Agentes que aprenden

- Aprendizaje automático: programas que mejoran su comportamiento con la experiencia (aprendizaje automático = software).
- Dos formas posibles de adquirir experiencia:
  - A partir de ejemplos suministrados por un usuario (ej. reconocimiento de patrones: un conjunto de ejemplos clasificados o etiquetados es la fuente de información o la experiencia necesaria para el aprendizaje). **APRENDIZAJE SUPERVISADO.**
  - Mediante exploración autónoma (ej. software que aprende a jugar al ajedrez mediante la realización de miles de partidas contra sí mismo; o robot que aprende a salir de un laberinto mediante prueba y error). **APRENDIZAJE NO SUPERVISADO.**
- El aprendizaje en agentes **puede definirse como el proceso de modificación de cada componente** del agente, lo cual permite mejorar el nivel de la actuación del agente.

# Mas características (no todas necesarias)

- Franklin y Nwana añaden nuevas características a los agentes:
- **Continuidad temporal:** el agente es un proceso sin ningún tipo de limitación temporal.
- **Autonomía:** el agente actúa sobre la base de su propio criterio y en la percepción que tiene de su entorno.
- **Racionalidad:** el agente hace siempre “lo correcto”, siempre según su criterio.
- **Adaptabilidad:** está relacionada con el aprendizaje que un agente es capaz de realizar y si puede cambiar su comportamiento basándose en ese aprendizaje.
- **Movilidad:** capacidad de un agente de trasladarse a través de una red telemática.
- **Veracidad:** se supone que un agente no comunica información falsa a propósito.
- **Benevolencia:** con esto lo que se observa es que un agente está dispuesto a ayudar a otros, siempre y cuando esto no entre en conflicto con sus objetivos.

# Agentes autónomos

- La **recopilación de información** consiste en llevar a cabo acciones que modifiquen las percepciones futuras:
  - Exploración inicial un medio desconocido
  - Almacenar datos sobre sus propias acciones que mejoren la selección posterior
- Un agente es **autónomo** cuando se apoya más en su propias percepciones que el conocimiento inicial con el que fue diseñado
- Así se **adaptan** al ambiente mediante algún tipo de **aprendizaje**
  - Ejemplo: una aspiradora limpia un edificio de oficinas y lleva un registro de la suciedad de cada habitación según la hora.
    - ❖ Esto le permite acudir a una habitación a la hora en que está sucia

# Ejemplos

- Coche autónomo
- Sistema de diagnóstico médico
- Análisis de imágenes por satélite
- Controlador de una refinería
- Tutor de inglés interactivo
- Etc ...

# Ambientes / Entornos

- El tipo de entorno determina en gran parte el diseño del agente
  - **Total / parcialmente observable:**
    - el agente requiere **memoria** (estado interno)
  - **Discreto / continuo:**
    - el agente puede no ser capaz de “enumerar” todos **los estados**
  - **Estocástico / determinístico:**
    - el agente tiene que estar preparado para **contingencias**
  - **Agente único / multiagente:**
    - el agente puede necesitar un comportamiento **aleatorio**



# Ambiente observable vs. parcialmente observable

- Un entorno es **observable** si un agente **puede obtener información completa, correcta y actualizada** sobre su estado.
- Los sensores de un agente le dan acceso al estado completo del entorno en cada instante de tiempo.
- Cuanto **más observable** sea un entorno, **más fácil es construir agentes** que pueden operar en el mismo.
- La mayoría de los entornos de la vida real, **no son accesibles**.

# Ambiente discreto vs. continuo

- Un entorno es discreto si hay un **número fijo, finito de acciones y percepciones** en el mismo. En este caso, el entorno puede quedar descrito por un número limitado de percepciones y acciones claramente definidas.
- El ajedrez describe un entorno discreto. La conducción de un taxi se encuentra en un entorno continuo (**puede suceder cualquier cosa**).
- Evidentemente, los entornos discretos son más fáciles para los desarrolladores.

# Ambiente determinista vs. no determinista

- Un entorno es **determinista** si cualquier acción tiene **un único** efecto sobre él, y **no** hay **incertidumbre** sobre el estado resultante. Se tiene disponible toda la información necesaria para la toma de decisión.
- El siguiente estado del entorno **está completamente determinado** por el estado actual y la **acción ejecutada** por el agente. Al ejecutar una acción, siempre se obtendrá el mismo resultado.
- Los entornos no deterministas son más problemáticos.
- El mundo físico, a todos los efectos, se puede considerar como no determinista.
- En entornos complejos, aunque sean esencialmente deterministas, predecir el efecto de una acción puede ser demasiado complejo para ser factible.
- Los modelos **estocásticos**, son aquellos en el que **no se tiene toda la información** con anticipación, incorporando así la incertidumbre.

# Ambiente episódico vs. no episódico

- Un entorno es **episódico** si el comportamiento del agente puede ser dividido en **secuencias de percepción-acción no relacionados entre sí** (episodios).
- Así, la experiencia del agente se divide en "episodios" atómicos (cada episodio consiste en el agente percibiendo, y después, realizando una sola acción), y **la elección de la acción en cada episodio sólo depende del propio episodio** (como en las clásicas cadenas de Markov).
- Los entornos episódicos son más fáciles para los desarrolladores porque el agente puede decidir qué acción realizar sólo sobre la base del episodio actual; no necesita recordar episodios anteriores o razonar sobre los próximos, **permitiendo** así al agente que **no se preocupe por la planificación de lo que pueda ocurrir**.

# Ambiente estático vs. dinámico

- Un ambiente es **estático** si éste no cambia mientras el agente está “pensando” (deliberando). No tiene importancia el tiempo que se usa en pensar y no necesita monitorizar el mundo mientras piensa. El tiempo carece de valor mientras se computa una buena estrategia. En otro caso será dinámico.
- Los entornos dinámicos son más difíciles para el desarrollador ya que otras entidades pueden interferir con las acciones del agente.
- Muchos entornos de la vida real son muy dinámicos:
  - El mundo real
  - Internet ...

# Sistemas multi-agente

## ➤ Diferencia entre un agente y multi-agente:

- **Agente:** utiliza el concepto de agente como mecanismo de abstracción pero podría ser implementado con estructuras software tradicionales.
- **Multi-agente:** se diseña e implementa pensando en que estará compuesto por varios agentes que interactuarán entre sí, de forma que juntos permitan alcanzar la funcionalidad deseada. En este caso se exige un esfuerzo de abstracción en el diseño, identificar mecanismos de aprendizaje, coordinación, negociación, etc.

## ➤ Características más relevantes de los sistemas multi-agentes:

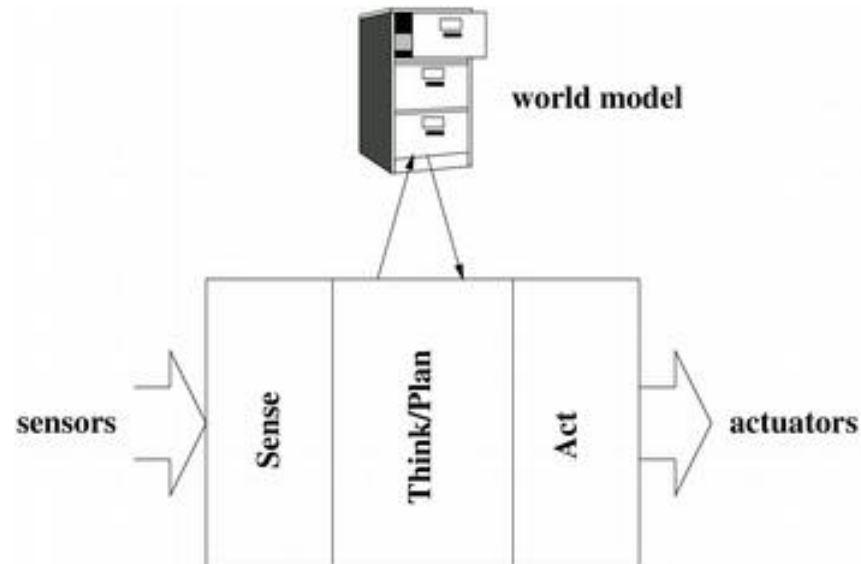
- **Cooperación:** generan deberes mutuamente dependientes en actividades conjuntas.
- **Resolución de conflictos:** debido a la coexistencia, pueden darse circunstancias que desemboquen en conflictos. Dichos conflictos son gestionados a través de **negociación**.
- **Negociación:** punto de encuentro que maximice la utilidad del sistema global. Zlotkin y Rosenschein (1990), Lesser (1988) y Sycara (1989):
  - Cada agente hace su propuesta hasta llegar a un plan conjunto o se crea un agente coordinador.
- **Compartición del conocimiento:** representación del conocimiento y protocolos de comunicación.

# Entornos de trabajo

| Entornos de trabajo                                     | Observable                   | Determinista                | Episódico                | Estático             | Discreto             | Agentes                  |
|---|------------------------------|-----------------------------|--------------------------|----------------------|----------------------|--------------------------|
| Crucigrama<br>Ajedrez con reloj                         | Totalmente<br>Totalmente     | Determinista<br>Estratégico | Secuencial<br>Secuencial | Estático<br>Semi     | Discreto<br>Discreto | Individual<br>Multi      |
| Póker<br>Backgammon                                     | Parcialmente<br>Totalmente   | Estratégico<br>Estocástico  | Secuencial<br>Secuencial | Estático<br>Estático | Discreto<br>Discreto | Multi<br>Multi           |
| Taxi circulando<br>Diagnóstico médico                   | Parcialmente<br>Parcialmente | Estocástico<br>Estocástico  | Secuencial<br>Secuencial | Dinámico<br>Dinámico | Continuo<br>Continuo | Multi<br>Individual      |
| Análisis de imagen<br>Robot clasificador                | Totalmente<br>Parcialmente   | Determinista<br>Estocástico | Episódico<br>Episódico   | Semi<br>Dinámico     | Continuo<br>Continuo | Individual<br>Individual |
| Controlador de refinería<br>Tutor interactivo de inglés | Parcialmente<br>Parcialmente | Estocástico<br>Estocástico  | Secuencial<br>Secuencial | Dinámico<br>Dinámico | Continuo<br>Discreto | Individual<br>Multi      |

# Arquitecturas

## ➤ Deliberativa o simbólica



## ➤ Primera arquitectura utilizada en la era de la Inteligencia Artificial.

- Esquema **secuencial y repetitiva**
- El modelo se **representa mediante símbolos y predicados**
- El **módulo central**, trata de tener todo controlado, realizando muchos cálculos para ello.

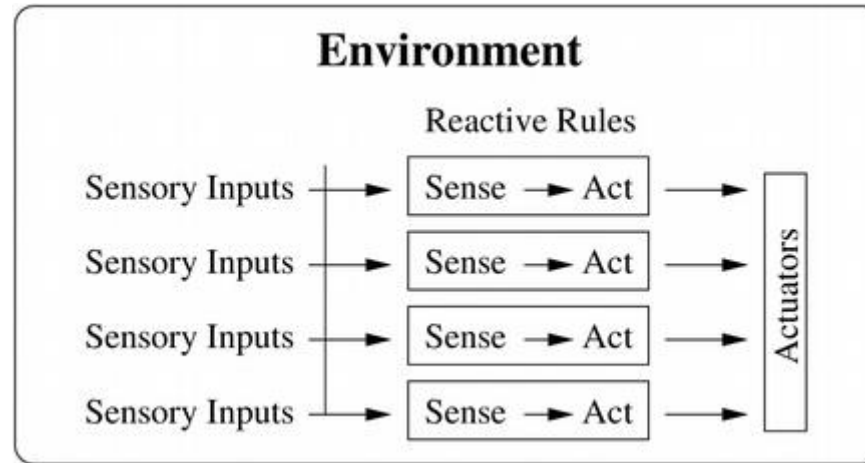
## ➤ Problemas:

- Al ser secuencial, si un módulo falla, la salida será incorrecta
- Representación del mundo muy compleja
- Incapaz de hacer frente a entornos dinámicos



# Arquitecturas multi-agente

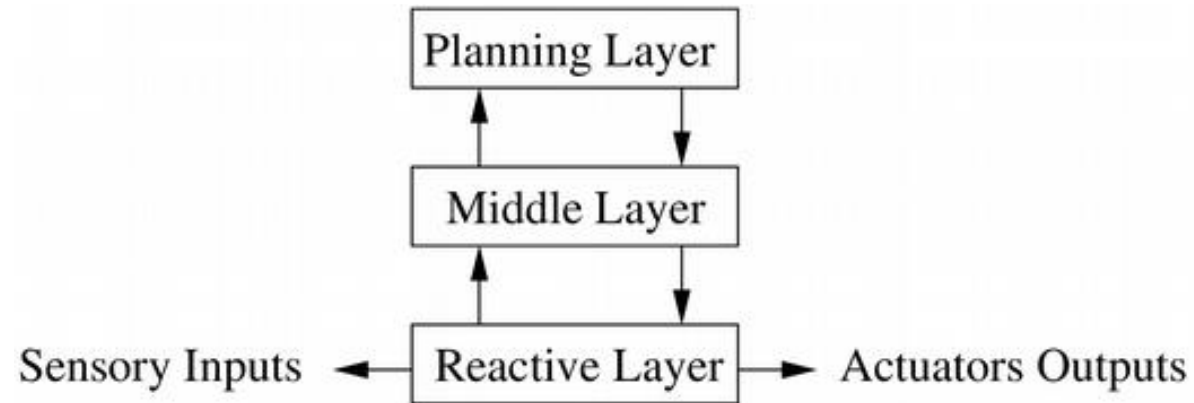
## ➤ Reactiva



- Una **arquitectura simple** que usa reglas predefinidas para las entradas sensoriales.
  - Mucho más rápida que la arquitectura **deliberativa**
  - No guarda la representación del entorno, pero sí el estado actual (interno).
  - **Compartición del conocimiento**: representación del conocimiento y protocolos comunicación.
- **Problemas**:
  - **No tiene memoria**, no mantiene ninguna representación interna ➡ **no puede aprender**.

# Arquitecturas multi-agente

## ➤ Híbrida



## ➤ Evolución de la arquitectura deliberativa.

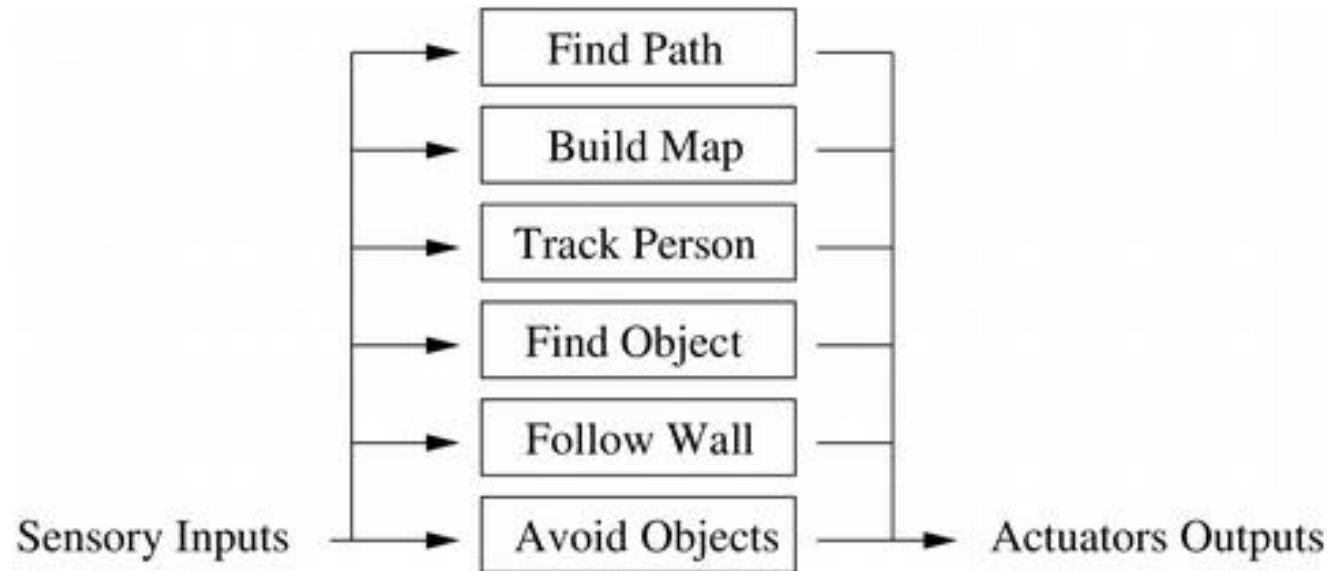
- Deliberativas ➡ son completas respecto a la inteligencia, pero lentas.
- Reactivas ➡ rápidas pero sin inteligencia
- Híbridas son una mezcla entre deliberativas y reactivas (capas reactiva y deliberativa)

## ➤ Problemas:

- La capa deliberativa hace que el problema de representar entornos dinámicos siga ahí.
- Difícil diseñar la capa intermedia y muchas veces no es reusable.

# Arquitecturas multi-agente

## ➤ Basada en comportamiento



## ➤ Módulos en paralelo, inspirado en sistemas biológicas.

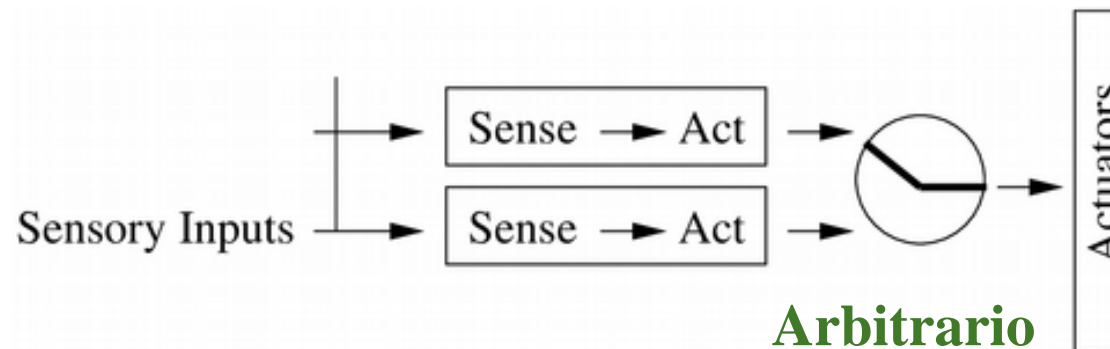
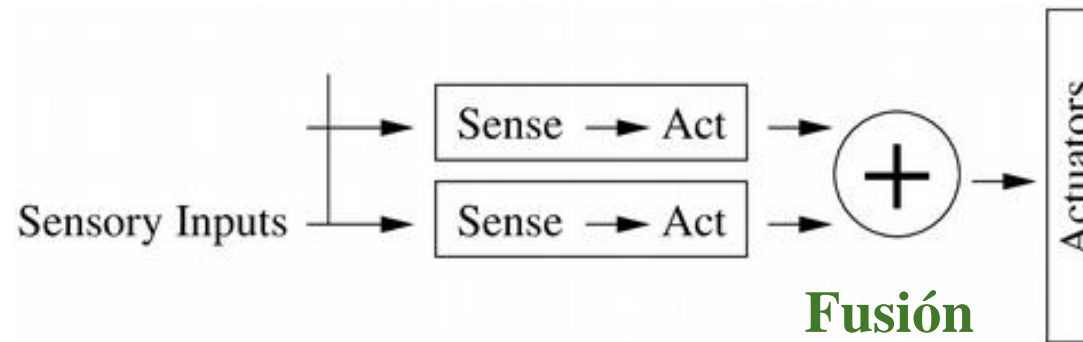
- Cada módulo (agente) tiene su propósito.
- Todos se ejecutan continuamente y se comunican entre ellos.

## ➤ Problemas:

- Los módulos deben de coordinarse ya que los actuadores solo pueden obedecer a un módulo cada vez.

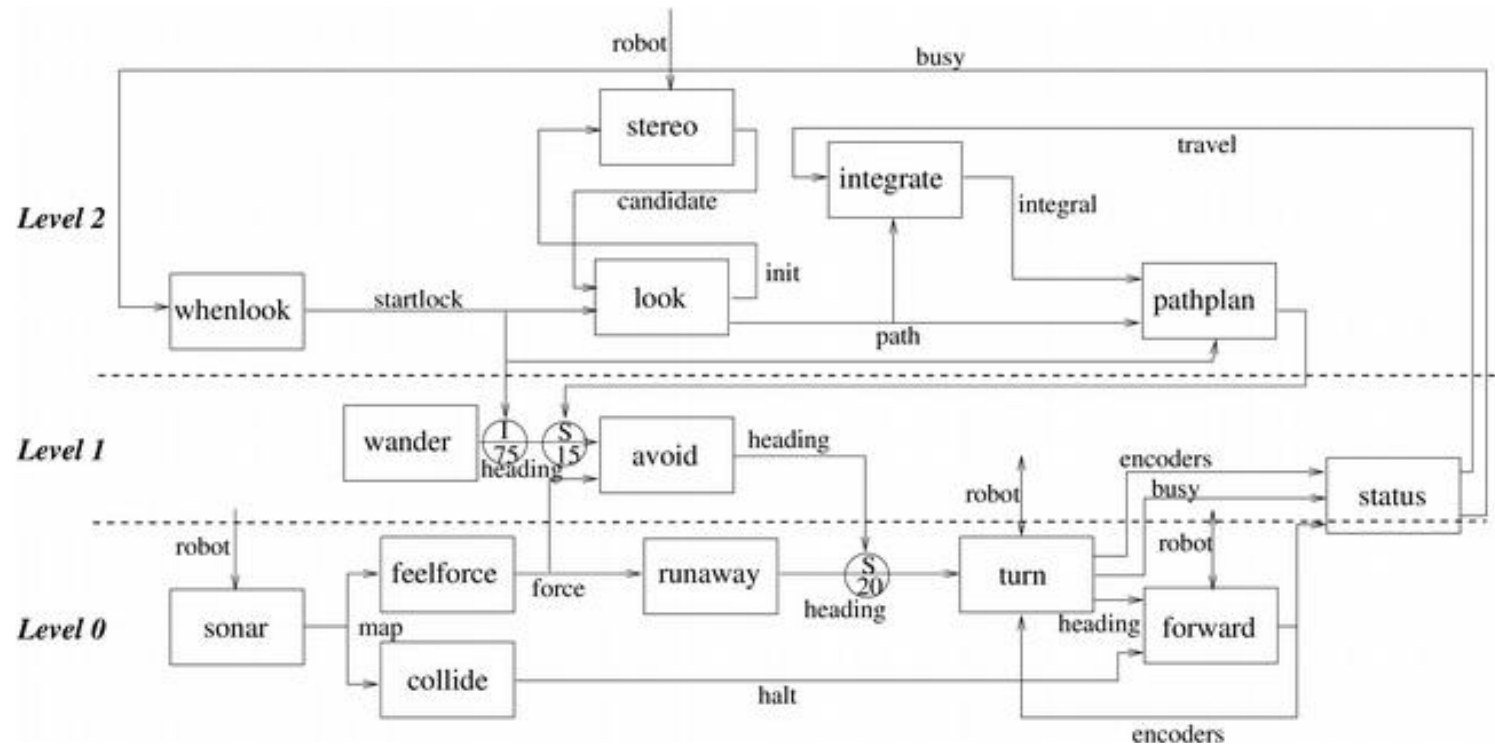
# Arquitecturas multi-agente

- Tipos de coordinación en arquitecturas basadas en comportamiento



# Arquitecturas multi-agente

## ➤ Subsunción



## ➤ Basada en la arquitectura de **comportamientos mediante capas**

- En cada capa puede haber más de un módulo, trabajando en paralelo.
- En cada capa se distribuyen diferentes comportamientos **ordenados por la complejidad**.
- Las capas superiores pueden inhibir las salidas de los módulos inferiores.

## ➤ Problemas: **Difícil diseñar la arquitectura.**

# Técnicas para implementar la racionalidad

- Representación del Conocimiento
- Razonamiento simbólico
- Planificación
- Satisfacción de Restricciones
- Aprendizaje Automático
- Razonamiento bajo incertidumbre:
  - Sistemas de mantenimiento de la verdad
  - Inferencia estadística