

Inteligencia Artificial

Agentes Inteligentes

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 **crear 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:
 - **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,

Objeto VS Agente

➤ Objetos:

- Los objetos tienen capacidad para el paso de mensajes a otros objetos
 - Ejemplo: **Java**
 - Métodos **privados y públicos**
 - El objeto conoce 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:**

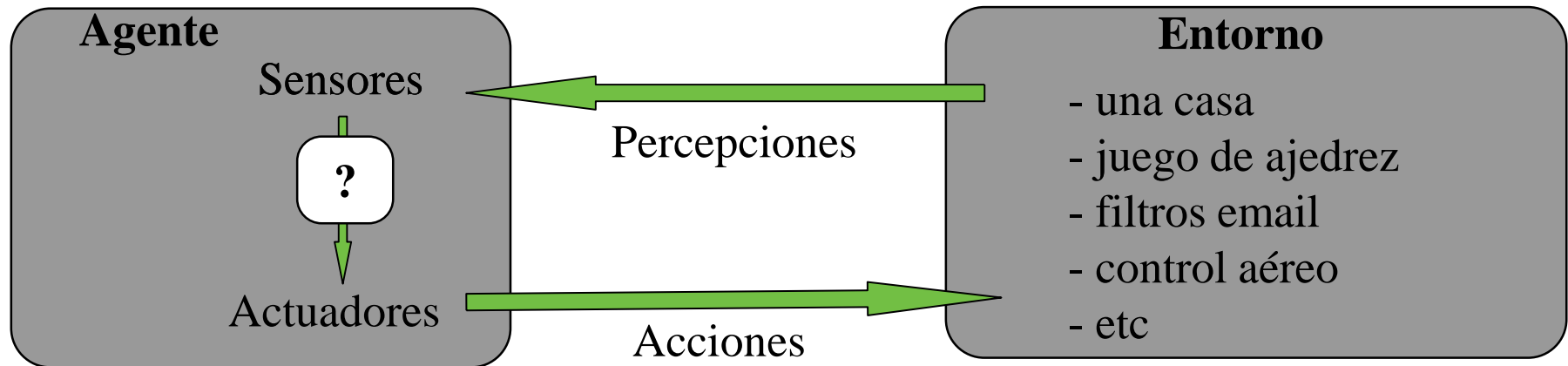
- Un agente autónomo es un sistema computacional que habita en un entorno dinámico complejo, percibiendo y actuando autónomamente en este entorno, y haciendo esto para realizar un conjunto de objetivos o tareas para los cuales fueron diseñados.

➤ **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

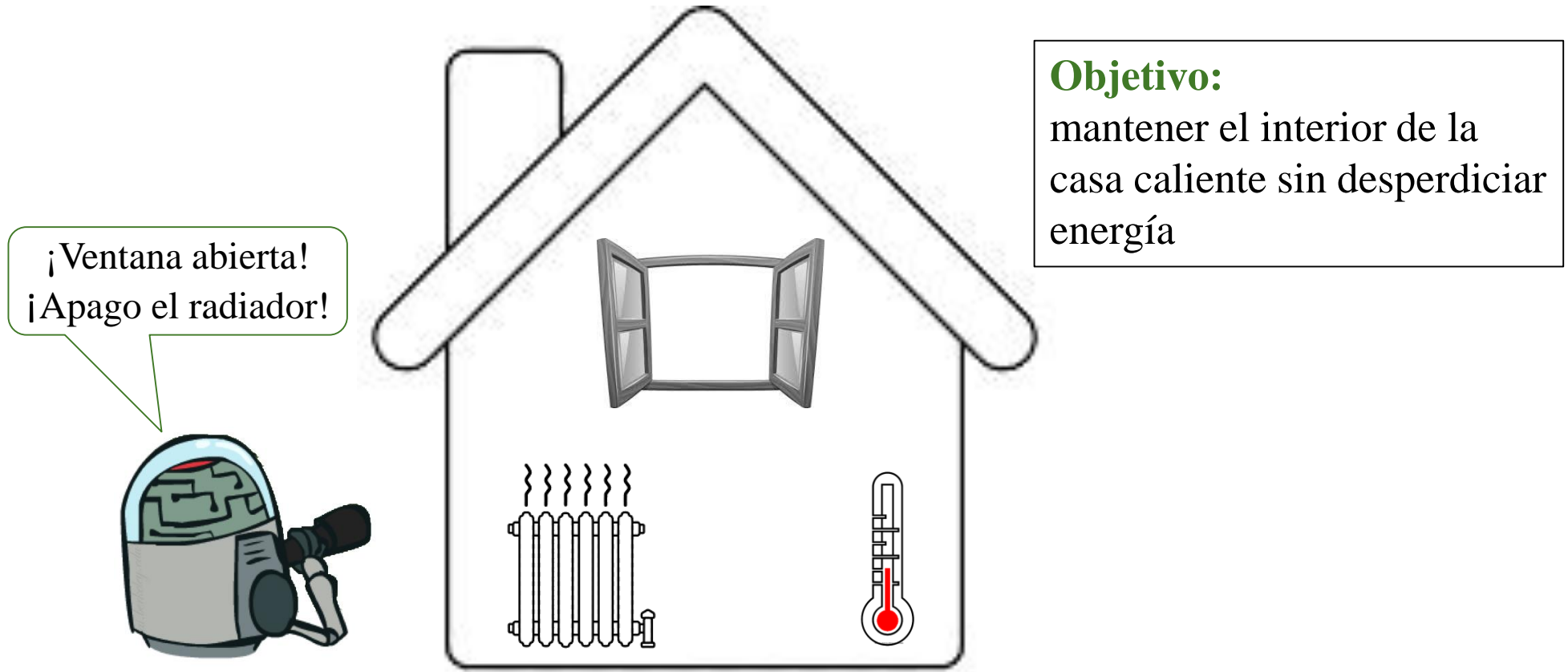
- Un agente **percibe su entorno** a través de sensores y **actúa sobre él** a través de **actuadores**



- 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?
 - **Autonomía**: **cuando se apoya más en su propias percepciones que el conocimiento inicial** (va aprendiendo)
 - **Flexibilidad**: **reactivo, pro-activo y comunicativo**

Reactividad (Reacción)

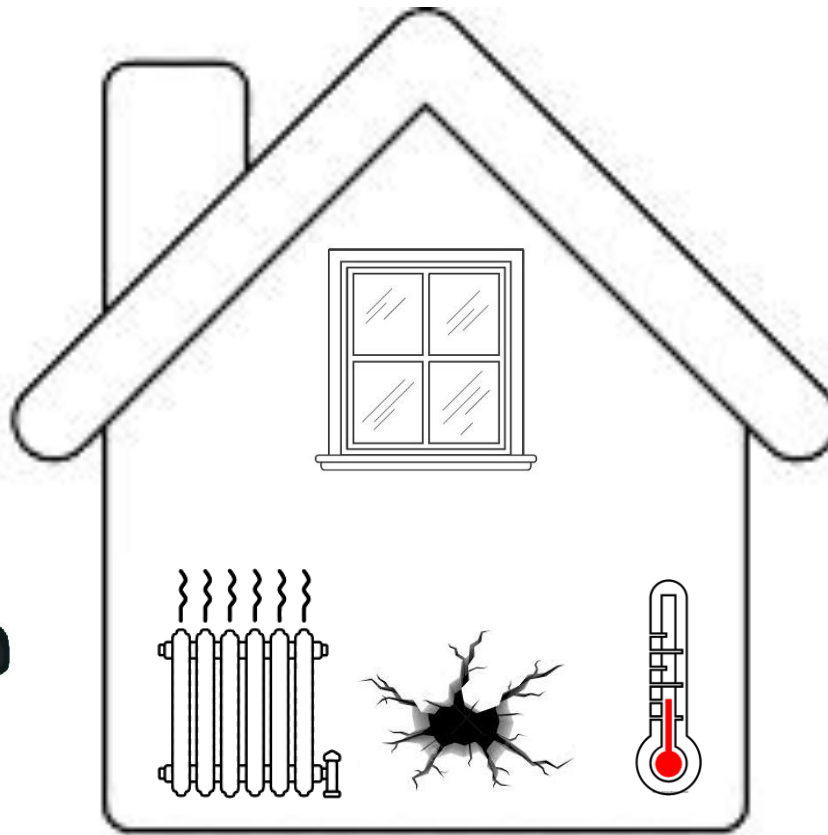
- Los agentes reactivos **no poseen modelos simbólicos de su entorno**; en su lugar **actúan y responden a los estímulos** que presenta el estado actual del entorno en el que están embebidos.



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!

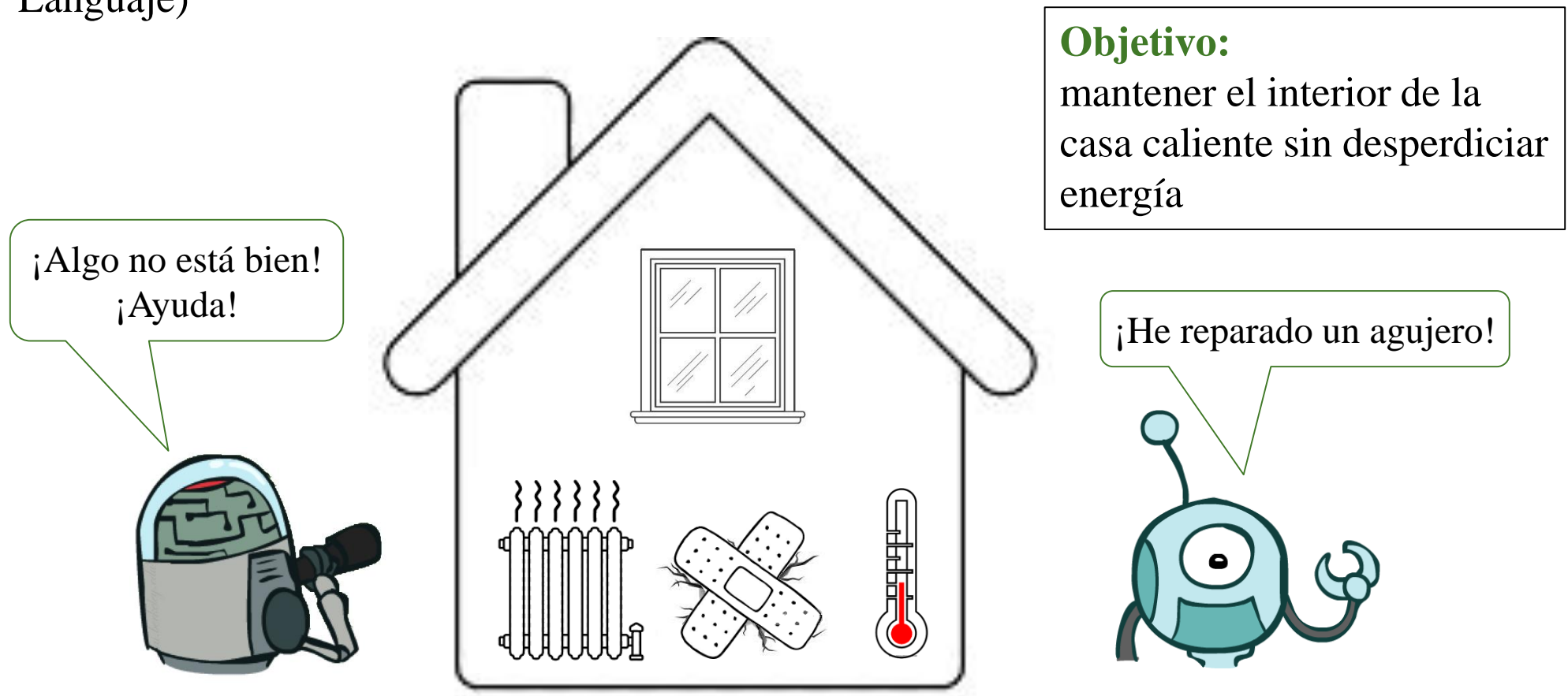


Objetivo:

mantener el interior de la casa caliente sin desperdiciar energía

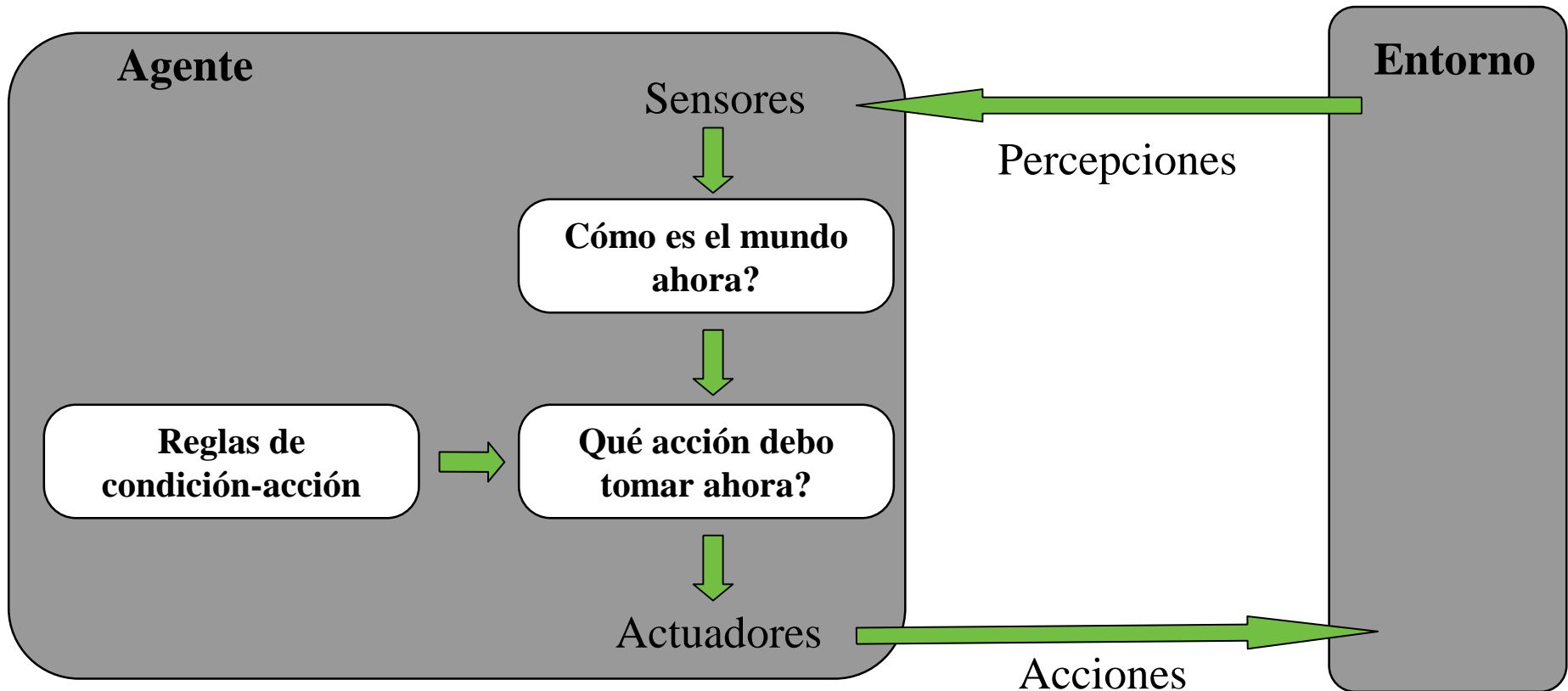
Comunicación

- 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.



- 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.

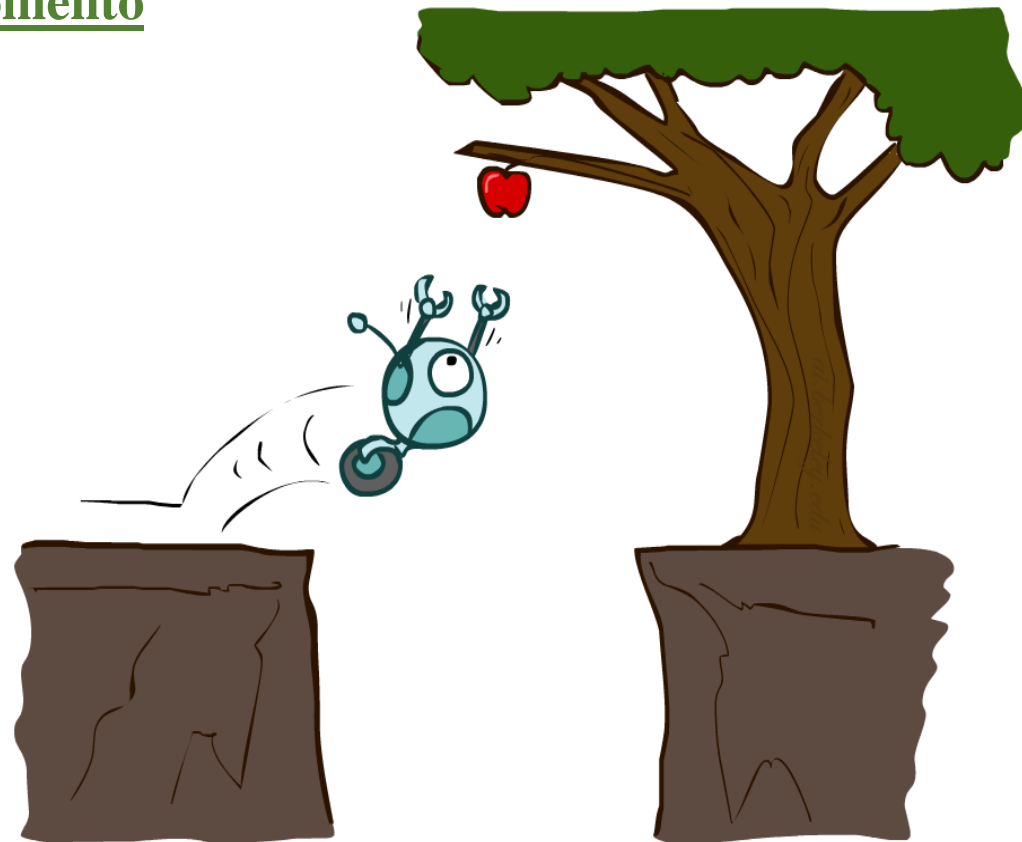
Sensor frontal	Sensor derecho	Sensor izquierdo	Acción
libre	libre	libre	avanzar
no libre	libre	libre	girar derecha
no libre	no libre	libre	girar izquierda
...

Agentes Reactivos (resumen)

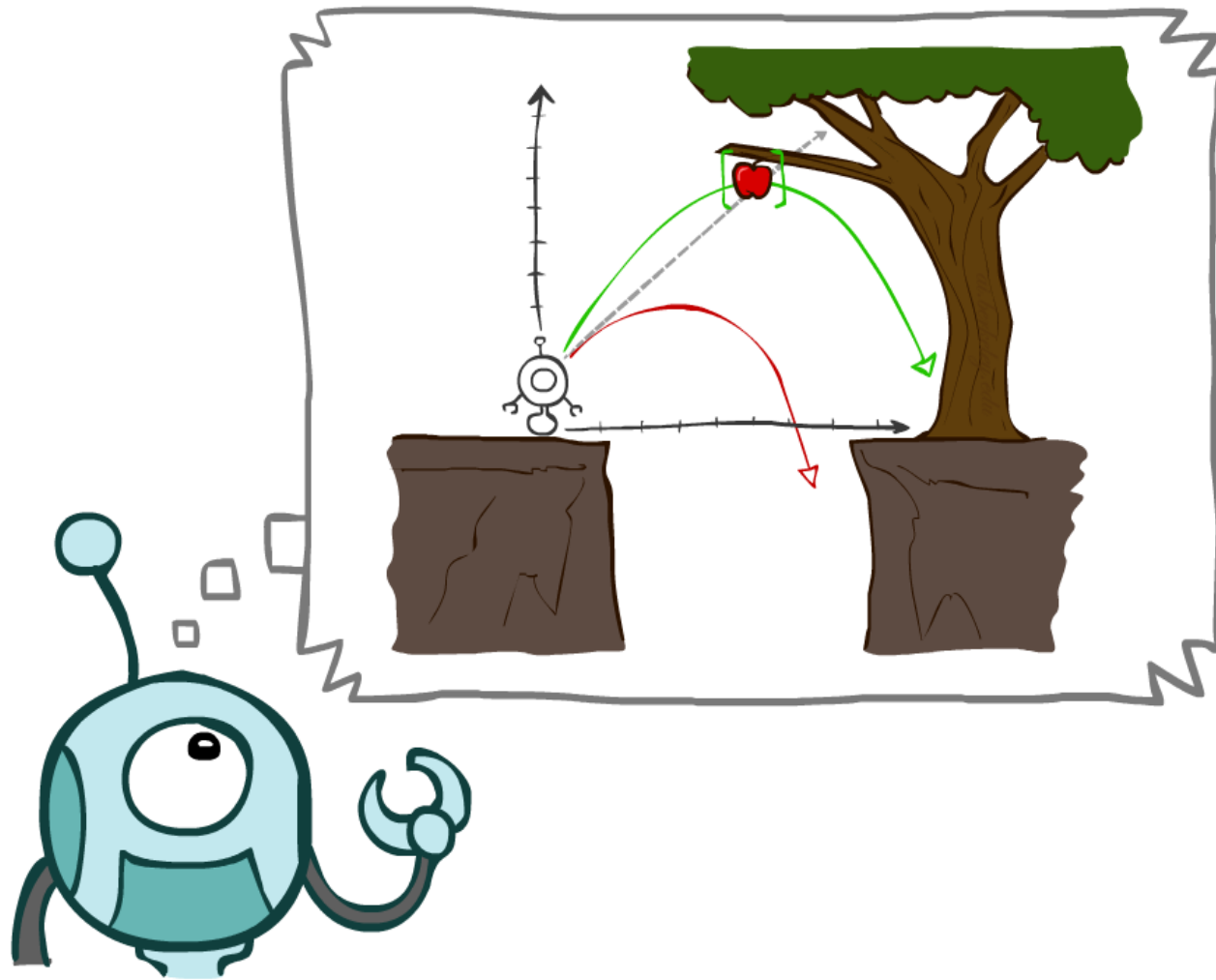
➤ Agentes Reactivos / Reflex:

- 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 reactivo ser **racional**?

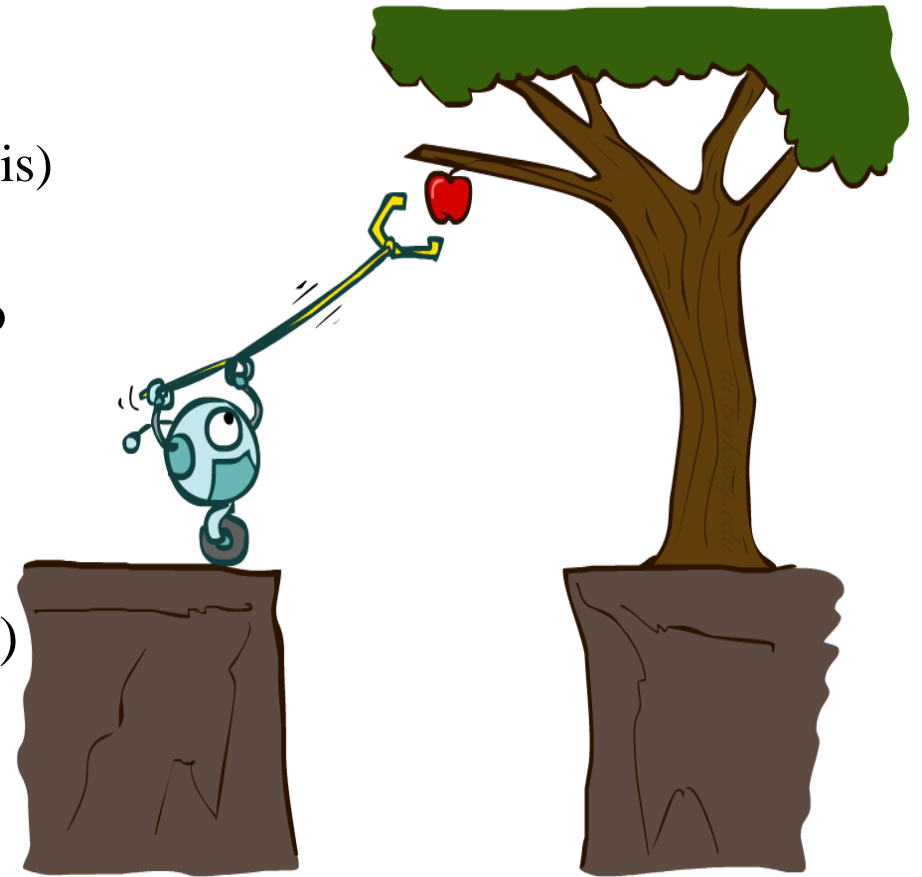


Agentes que planifican



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** (**tema 3**)



Tres tipos de agentes que planifican

➤ Agentes **con memoria (modelos)**

- con un estado interno que sirve para seguir la pista a los estados pasados de su mundo.

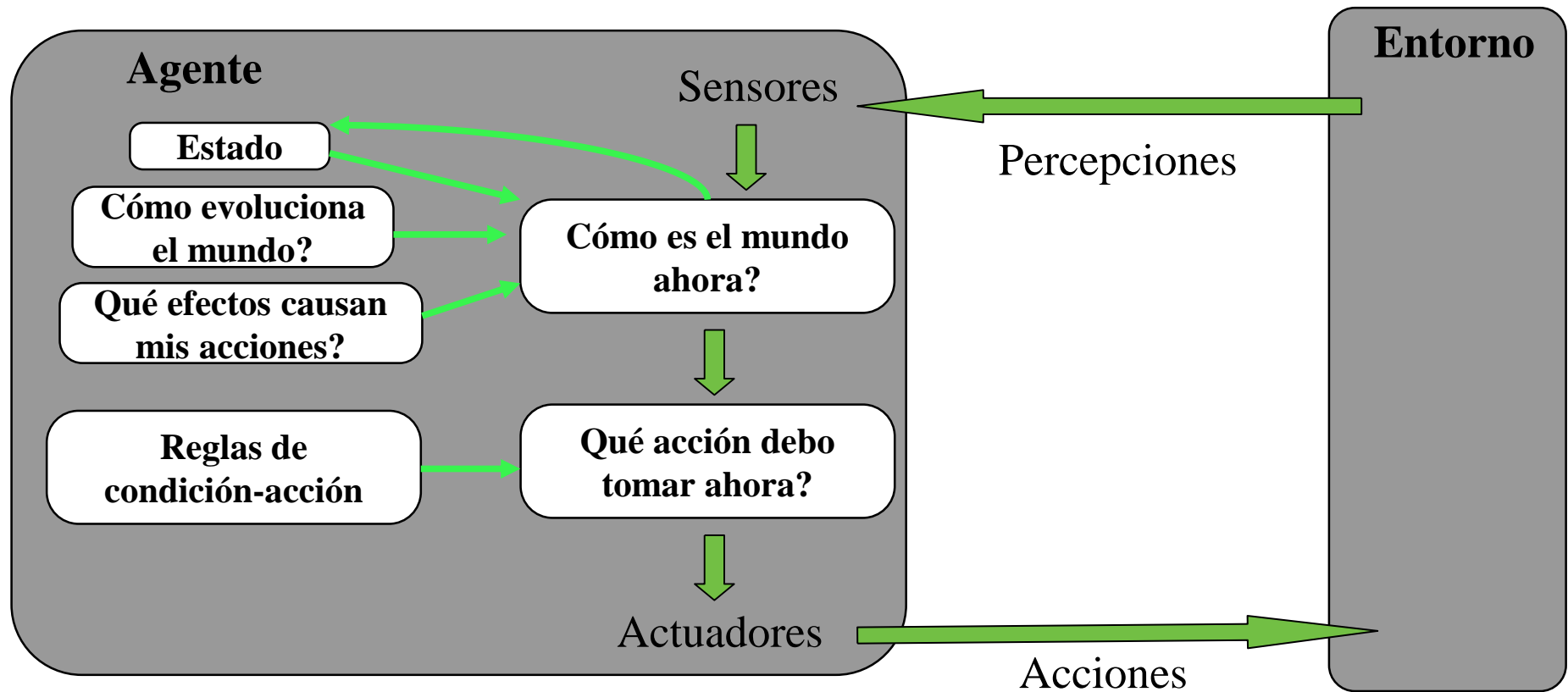
➤ Agentes **focalizados en encontrar la meta**

- además de disponer del modelo, tienen información sobre situaciones deseables para alcanzar la meta.

➤ Agentes **focalizados en mejorar la utilidad de sus acciones**

- Disponen de todo lo anterior y basan su decisión en 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 el objetivo 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**.

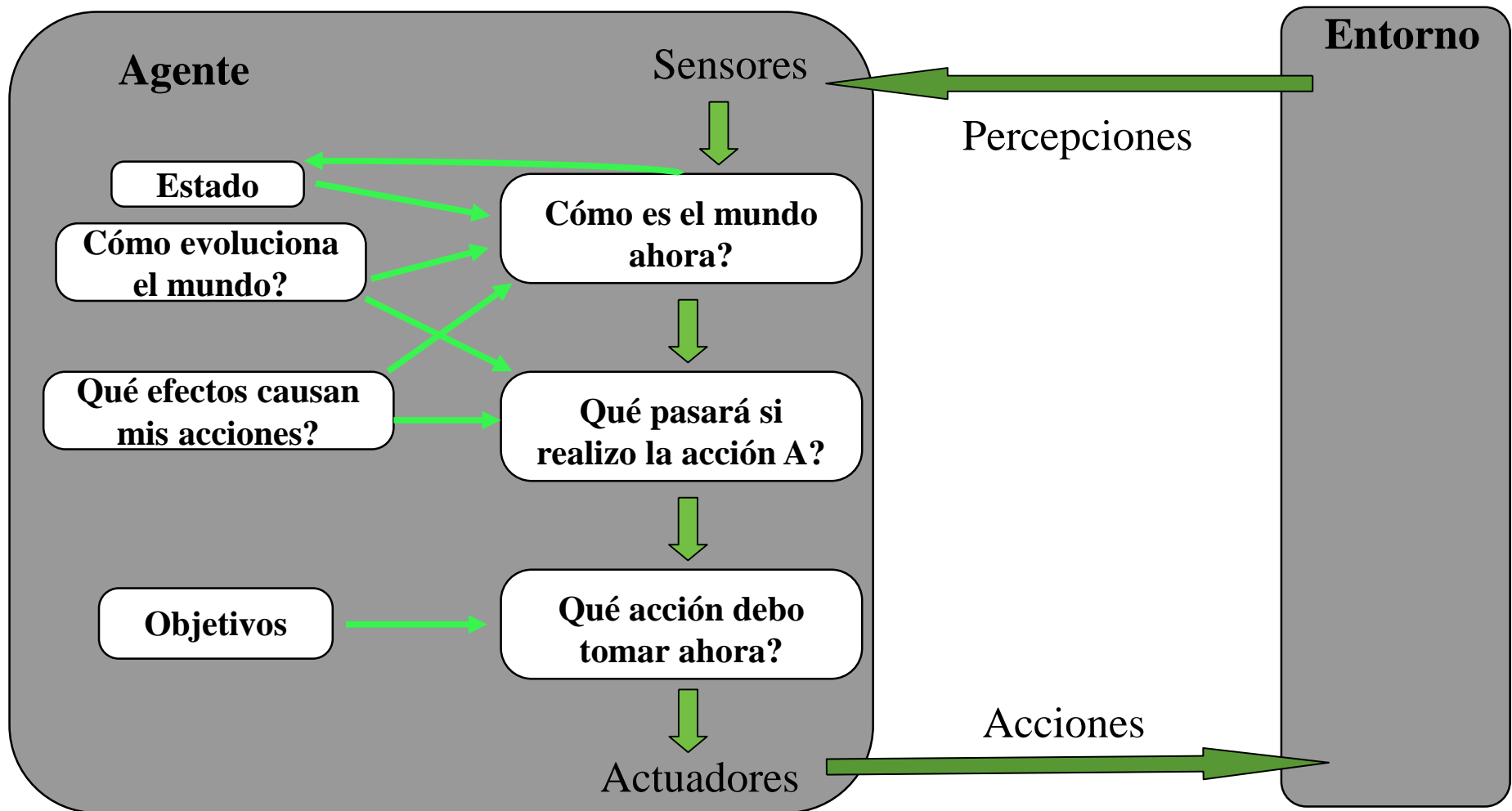
➤ El comportamiento se puede representar mediante una tabla

- 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, evitar bucles).

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
```

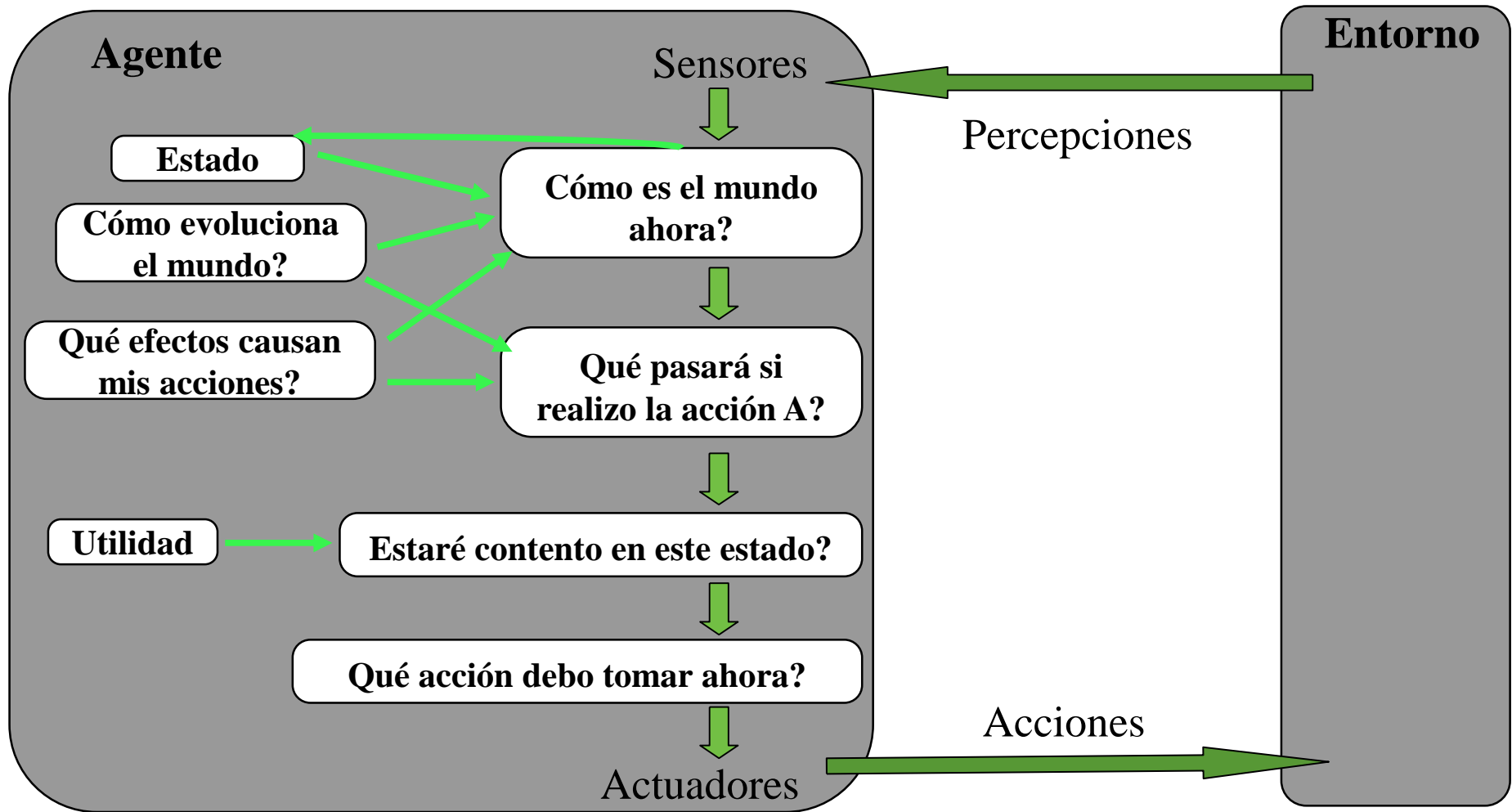
➤ Razonando acerca de acciones:

- 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**

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 **objetivo** o **situación final** a alcanzar por el robot (**objetivo**).
- Un **elemento** que **permite conocer el efecto de cada acción**. Por ejemplo, **¿estoy más cerca?**
 - El robot busca **en cada estado una acción que permita llegar al punto de destino** (o acercarse a él).
- Cómo escoger la mejor regla? Cómo argumentarlo?
 - hay que **maximizar la utilidad de las acciones**: elegir aquella que logra la mejor meta de todas.

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**

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 **objetivo** o **situación final a alcanzar** por el robot.
- Un **elemento** que **permite conocer el efecto de cada acción**.
- **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 basados en objetivos vs utilidad

- 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**
- **Utilidad:**
 - Función que caracteriza el grado de satisfacción (**PRINCIPIO DE MÁXIMA UTILIDAD ESPERADA**)

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.
- **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 aún más racionales: pueden aprender

- 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 permite mejorar el nivel de la actuación del agente.**

Ejemplos

- Coche autónomo
- Sistema de diagnóstico médico
- Análisis de imágenes por satélite
- Controlador de una refinería
- 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**
 - **Episódico / secuencial:**
 - el agente tiene que estar preparado para **planificar**
 - **Estático / dinámico**
 - **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**.
- En **entornos parcialmente observables** el agente requiere **memoria** (modelo).

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 estocásticos

- 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 modelos **estocásticos**, son aquellos en el que no se tiene toda la información con anticipación, **incorporando así la incertidumbre**.

Ambiente episódico vs secuencial

- Los **entornos episódicos** 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**.
 - Por ejemplo, un agente que tenga que seleccionar partes defectuosas en una **cadena de montaje** basa sus decisiones en la parte que está evaluando en cada momento, sin tener en cuenta decisiones previas; más aún, a la decisión presente no le afecta el que la próxima fase sea defectuosa.
- En ambientes **secuenciales**, las **percepciones-acciones tienen relación con las anteriores y las posteriores**.
 - Por ejemplo, el juego del **ajedrez**: las acciones que se realizan a corto plazo pueden tener consecuencias a largo plazo.

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.

Sistemas multi-agente

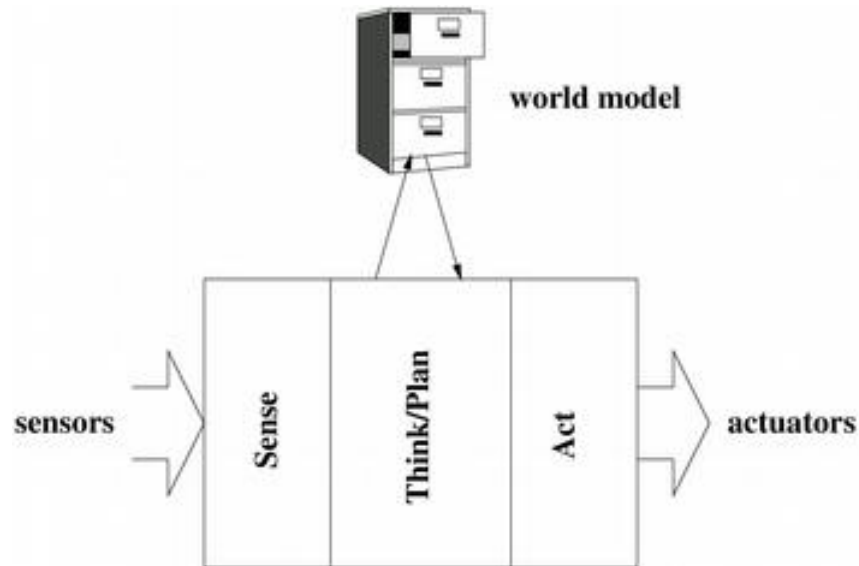
- **Multi-agente:** se diseña e implementa pensando en que **estará compuesto por varios agentes que interactuarán entre sí**, de forma que juntos permitan alcanzar la funcionalidad deseada.
- **Características más relevantes** de los sistemas multi-agentes:
 - **Cooperación:** generan deberes mutuamente 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 Ajedrez con reloj	Totalmente Totalmente	Determinista Estratégico	Secuencial Secuencial	Estático Semi	Discreto Discreto	Individual Multi
Póker Backgammon	Parcialmente Totalmente	Estratégico Estocástico	Secuencial Secuencial	Estático Estático	Discreto Discreto	Multi Multi
Taxi circulando Diagnóstico médico	Parcialmente Parcialmente	Estocástico Estocástico	Secuencial Secuencial	Dinámico Dinámico	Continuo Continuo	Multi Individual
Análisis de imagen Robot clasificador	Totalmente Parcialmente	Determinista Estocástico	Episódico Episódico	Semi Dinámico	Continuo Continuo	Individual Individual
Controlador de refinería Tutor interactivo de inglés	Parcialmente Parcialmente	Estocástico Estocástico	Secuencial Secuencial	Dinámico Dinámico	Continuo Discreto	Individual Multi

Arquitecturas

➤ Deliberativa o simbólica



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

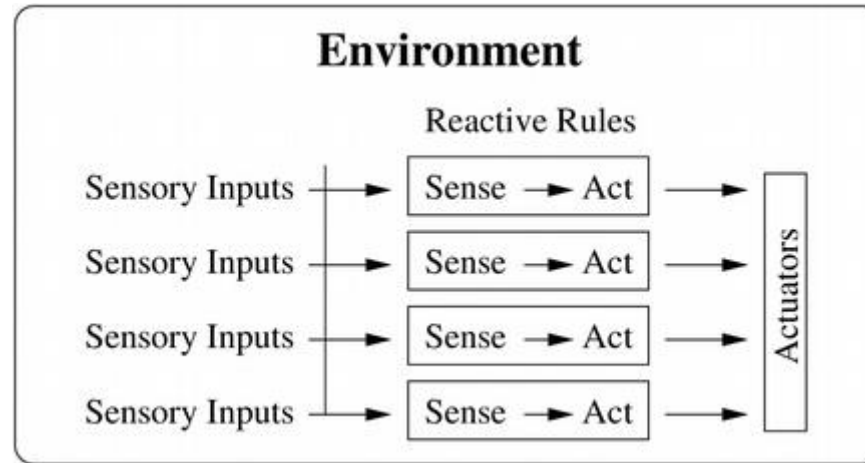
- Esquema **secuencial y repetitiva**
- El modelo se **representa mediante símbolos y predicados** (abstracto)
- El **módulo central**, trata de tener todo controlado, realizando muchos cálculos para ello.

➤ 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.

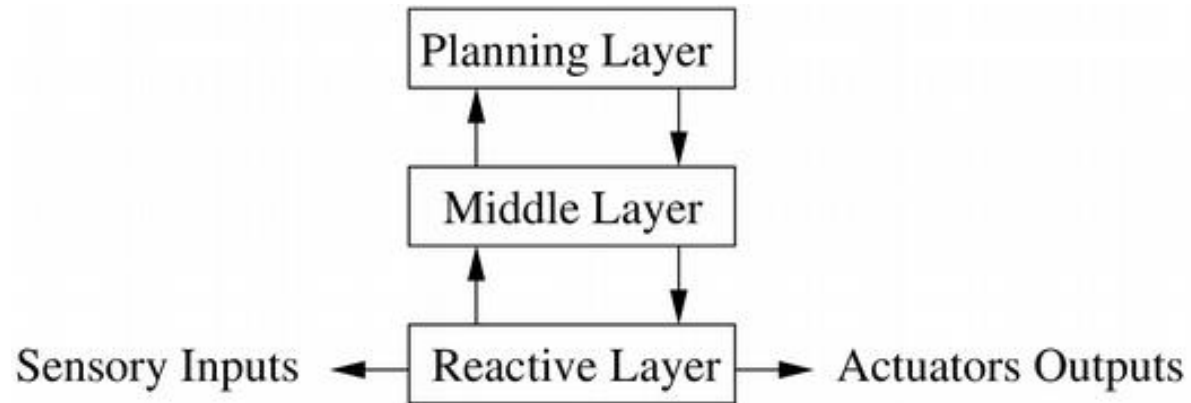
- Mucho más rápida que la arquitectura **deliberativa**
- No guarda la **representación** del entorno, pero sí el **estado actual** (interno).

➤ Problemas:

- **No tiene modelo**, no mantiene ninguna representación interna ➡ **no puede planificar**.

Arquitecturas multi-agente

➤ Híbrida



➤ Evolución de la arquitectura **deliberativa**.

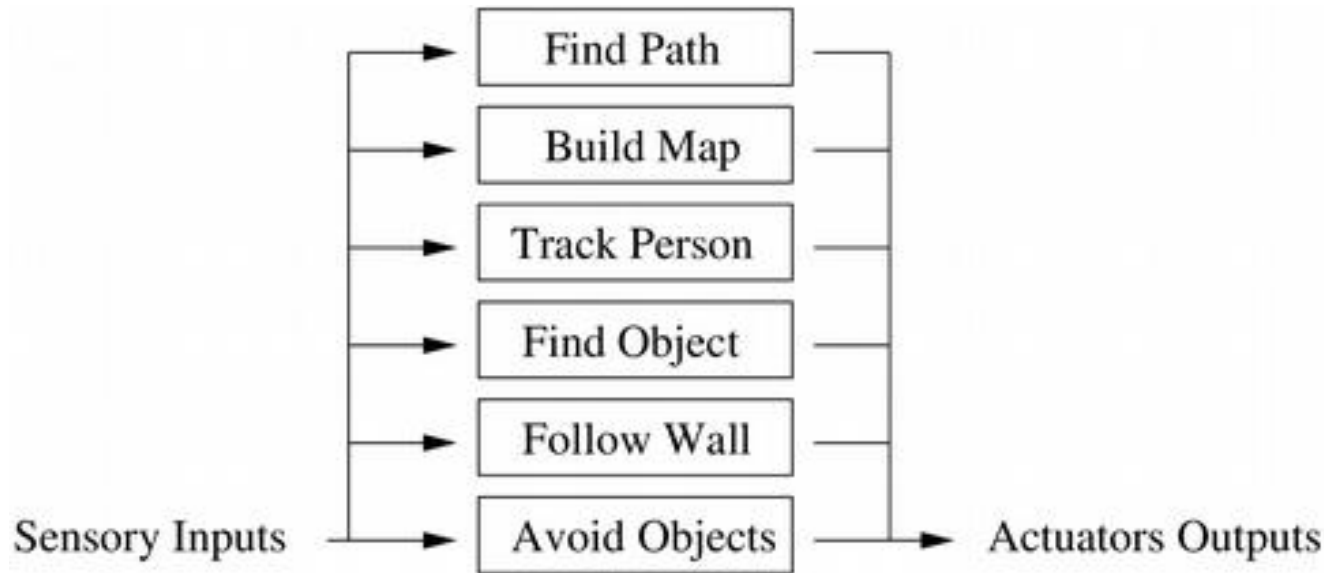
- Deliberativas ➡ son completas respecto a la inteligencia, **pero lentas**.
- Reactivas ➡ rápidas pero sin inteligencia
- Híbridas son una combinación entre deliberativas y reactivas (capas reactiva y deliberativa)

➤ 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.

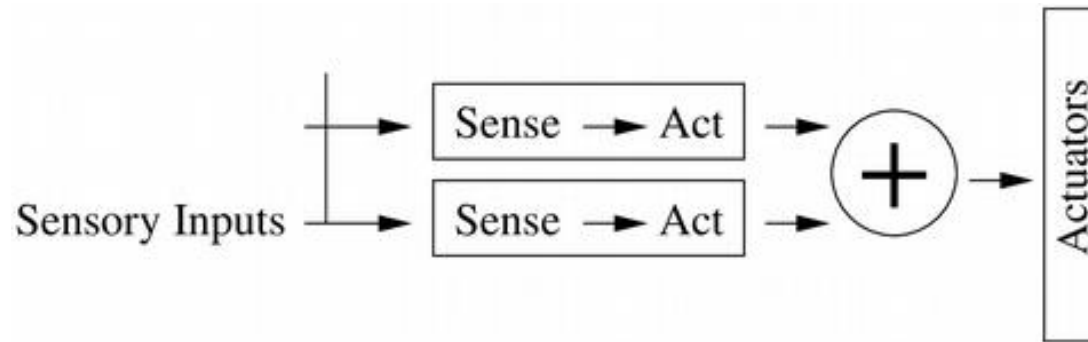
- Cada módulo (agente) tiene su propósito.
- Todos se ejecutan continuamente y se comunican entre ellos.

➤ Problemas:

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

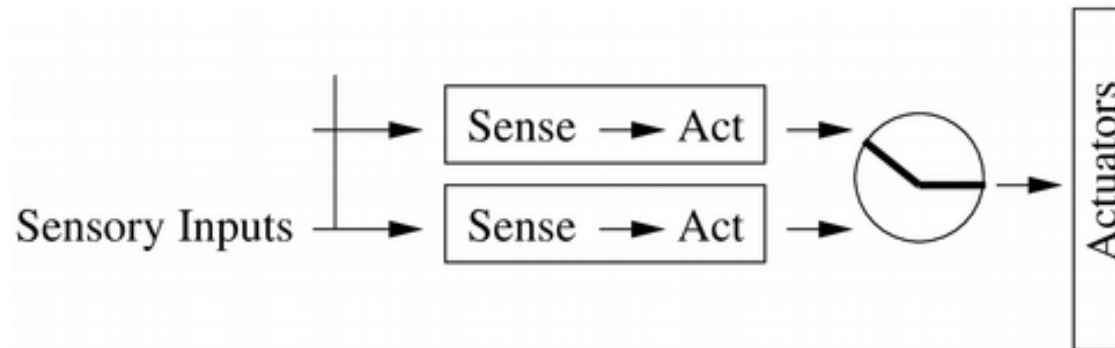
Arquitecturas multi-agente

➤ Tipos de coordinación en arquitecturas basadas en comportamiento



Fusión:

- Suma ponderada
- Combinación

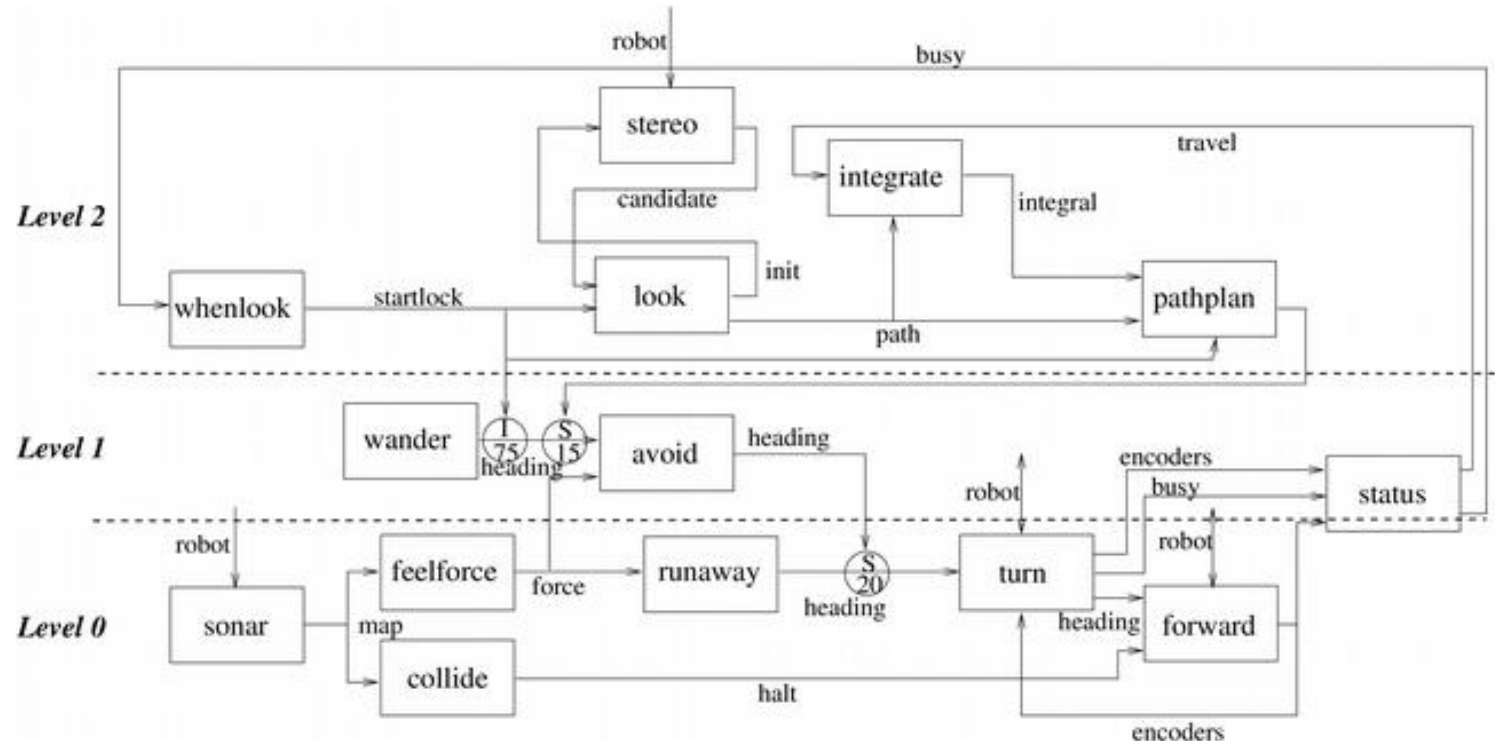


Arbitrario:

- Prioridad
- Votación

Arquitecturas multi-agente

➤ Subsunción



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

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

➤ Problema: **Difícil diseñar la arquitectura.**