



# Tema 4: Agentes Inteligentes

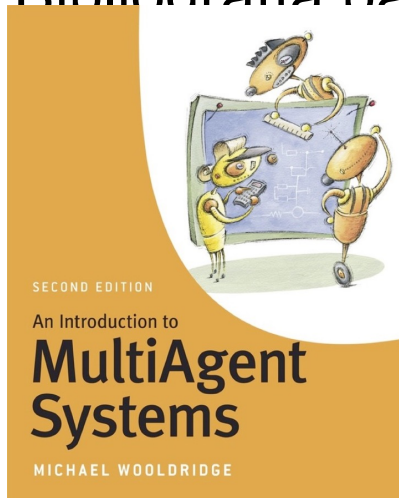


## Agentes inteligentes

- **Modelos y arquitecturas de agentes inteligentes**
- JASON: ejemplo de programa de agente
- Representación del conocimiento. Ontologías
- Comunicación y Sistemas multiagente



## Bibliografía básica:



### Tema 2

An Introduction to MultiAgent Systems – Second Edition

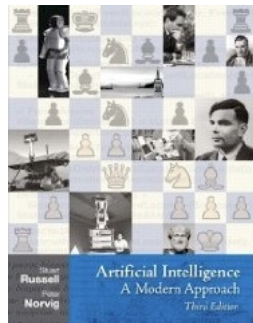
by Michael Wooldridge

Published May 2009 by John Wiley & Sons

ISBN-10: 0470519460

ISBN-13: 978-0470519462

## Bibliografía complementaria:



Artificial Intelligence: A Modern Approach (Third edition)

By Stuart Russell and Peter Norvig

Published by Prentice Hall Copyright © 2010

Published Date: Dec 1, 2009

ISBN-10: 0-13-604259-7

ISBN-13: 978-0-13-604259-4



# Definición de Agente



Los sistemas compuestos de múltiples agentes, comenzaron a utilizarse en la **Inteligencia Artificial Distribuida** (O'hare et al., 1996), tradicionalmente dividida en dos campos:

○ *Resolución de Problemas distribuidos:*

- Soluciones basadas en un cjto. de nodos cooperando en dividir y compartir.
- Cada agente tiene unas tareas y conducta prefijadas.
- El Sistema se centra en el comportamiento global.

○ *Sistemas Multiagente:*

- Agentes autónomos trabajan juntos para resolver problemas.
  - Cada agente tiene una información o capacidad incompleta para solucionar el problema.
  - Los agentes pueden decidir dinámicam. qué tareas realizar y quién realiza cada tarea.
- No hay un sistema global de control, datos descentralizados y computación asíncrona.

La investigación inicial progresa hacia la madurez. Surge la **Programación Orientada a Agentes** y los **Lenguajes de Comunicación de Agentes**

Posteriormente se plantea la **Ingeniería del Sw Orientada a Agente**.



## Definición de agente software

Varios tipos de definiciones

- Definiciones cualitativas

- Basadas en propiedades o en habilidades del agente, e.g. definición de Wooldridge & Jennings

- Definiciones operacionales

- Test de Huhns-Singh

"Un sist. que contiene 1 ó más ags. debería cambiar sustancialm. si otro ag. es añadido"

- Clasificaciones





# Definiciones

Agente (Diccionario RAE):

- Que obra o tiene virtud de obrar
  - El que realiza una acción
  - Persona o cosa que produce un efecto
- Persona que obra con poder de otra
  - El que actúa en representación de otro (agente artístico, comercial, inmobiliario, de seguros, de bolsa, etc)
- Persona que tiene a su cargo una agencia para gestionar asuntos ajenos o prestar determinados servicios

Agentes software:

- Aplicaciones inform. con capacidad para *decidir* cómo deben actuar para alcanzar sus objetivos

Agentes inteligentes:

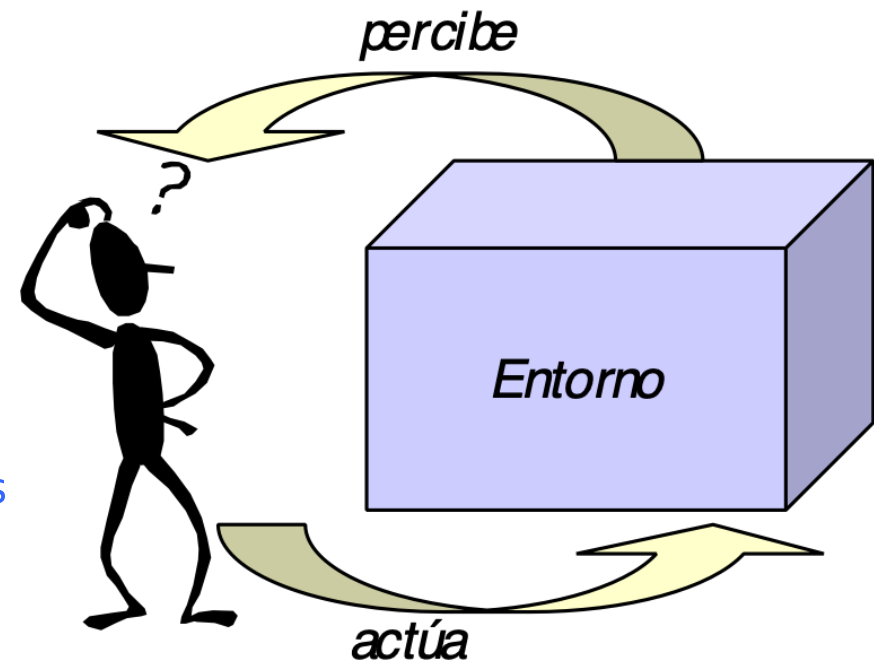
- Agentes Sw que pueden funcionar fiablem. en un entorno rápidam. cambiante e impredecible



La principal característica de los agentes es que son **autónomos**: capaces de actuar independientemente.

Una primera definición (Wooldridge):

Un agente es un sistema informático capaz de actuar autónomamente en algún entorno con el fin de alcanzar los objetivos que se le han delegado.



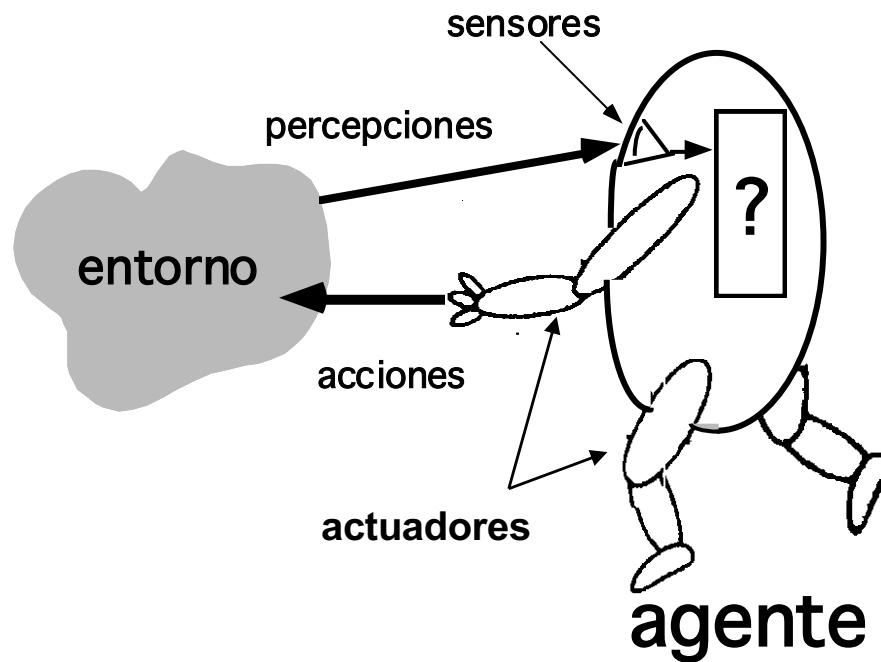
Consideramos que un agente como una entidad que está en continua interacción con su entorno:

percibe – decide – actúa – percibe – decide  
- .....

7







**Autónomo**  
**Reactivo**  
**Proactivo**

Razonamiento de alto nivel

**COGNICIÓN**

conducta  
dirigida  
por el objetivo

Razonamiento de bajo nivel

**ACCIÓN**

**PERCEPCIÓN**

actuadores que  
cambian el entorno

sensores que  
perciben  
el entorno



- Un termostato
  - objetivo delegado: mantener la temp. de una habitación.
  - acciones: calor/frio, conectar/desconectar.
- Programa biff UNIX
  - objetivo delegado: monitorizar correo entrante y etiquetarlo.
  - acciones: acciones GUI.
- Estos agentes son triviales porque las decisiones que toman (sus procesos cognitivos) son triviales



Houston, we've got a problem!!!

- ¿Proceso de larga vida (permanente)?
- ¿Independencia, autonomía?
- ¿"Inteligencia"?





[Franklin&Graesser, 96]



## The MuBot agent

*"The term agent is used to represent two orthogonal concepts. The first is the agent's ability for **autonomous execution**. The second is the agent's ability to perform **domain oriented reasoning**"*  
*[Virhagriswaran]*



## The Maes Agent [Maes, 95]

*"Autonomous agents are computational **systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed**"*





## The KidSim Agent [Smith et al., 94]

*"Let us define an agent as a **persistent software** entity dedicated to a specific purpose. 'Persistent' distinguishes agents from subroutines; agents have their **own ideas** about how to accomplish tasks, their **own agendas**. '**Special purpose**' distinguishes them from entire multifunction applications; agents are typically much smaller"*



## The Hayes-Roth Agent [Hayes-Roth, 95]

*"Intelligent agents continuously perform three functions: **perception** of dynamic conditions in the environment; **action** to affect conditions in the environment; and **reasoning** to interpret perceptions, **solve** problems, **draw** inferences, and **determine** actions"*



## The IBM Agent

*"Intelligent agents are **software entities** that carry out some set of operations **on behalf of a user** or another program with some degree of **independence or autonomy**, and in so doing, employ some **knowledge** or representation of the user's **goals or desires**"*

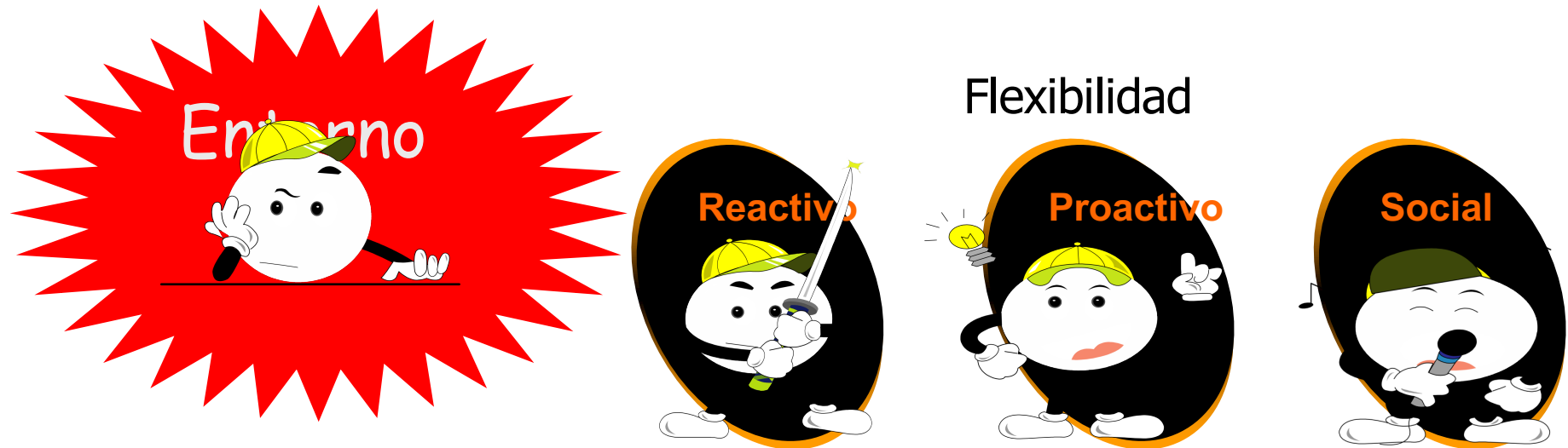


Un agente es un sistema informático, **situado** en algún entorno, que **percibe** el entorno (entradas sensibles de su entorno) y a partir de tales percepciones **determina** (mediante técnicas de resolución de problemas) y **ejecuta acciones** (de forma **autónoma** y **flexible**) que le permiten alcanzar sus **objetivos** y que pueden **cambiar** el entorno.

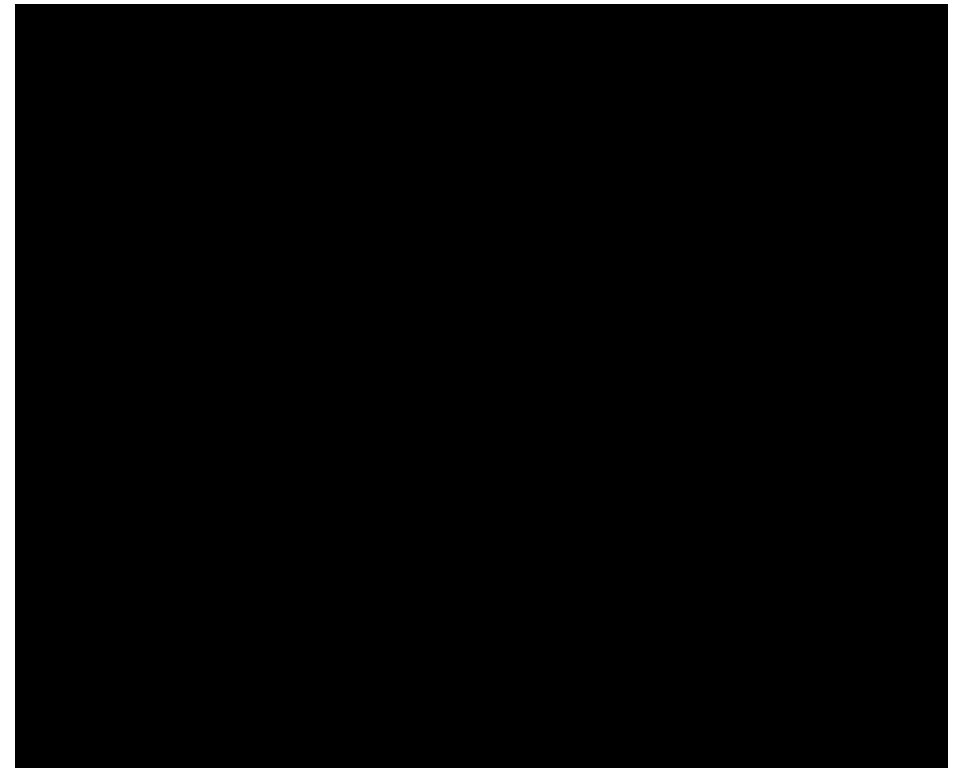


## ❖ Wooldridge:

Cualquier proceso computacional dirigido por el objetivo capaz de interaccionar con su entorno de forma **flexible** y **robusta**



- Si el entorno de un programa no cambia (es **estático**), un programa se puede ejecutar a ciegas.
- Mundo real **no estático**, la mayoría de los entornos son **dinámicos**.
- Programas para entornos dinámicos difíciles de construir: deben tener en cuenta la posib. de fallo, preguntarse si conviene continuar su ejecución.
- Sistema **reactivo** mantiene una constante interacción con su entorno y responde (a tiempo para que la respuesta sea útil) a los cambios que ocurren en él.





- Reaccionar a un entorno es fácil (por ejemplo, reglas **estimulo → respuesta**).
- Pero nosotros queremos que los agentes **hagan cosas por nosotros**.
- Por ello adoptamos un **comportamiento dirigido por el objetivo**.
- La **proactividad (iniciativa)** = generar e intentar conseguir objetivos, no dirigido solamente por eventos, tomar la iniciativa.
- Reconociendo oportunidades.



- El mundo real es un mundo multi-agente: cuando queremos conseguir objetivos tenemos que tener en cuenta al resto de entidades (agentes) del entorno donde nos encontramos.
- En algunos casos para alcanzar un objetivo es necesario interactuar con otros ya que puede que nosotros no tengamos capacidades suficientes para lograrlo.
- Del mismo modo que sucede en entornos de computadores: un testimonio es internet.
- *La sociabilidad (capacidad social)* en agentes es la capacidad de interactuar con otros agentes (y posiblemente con humanos) mediante *cooperación, coordinación y negociación*.

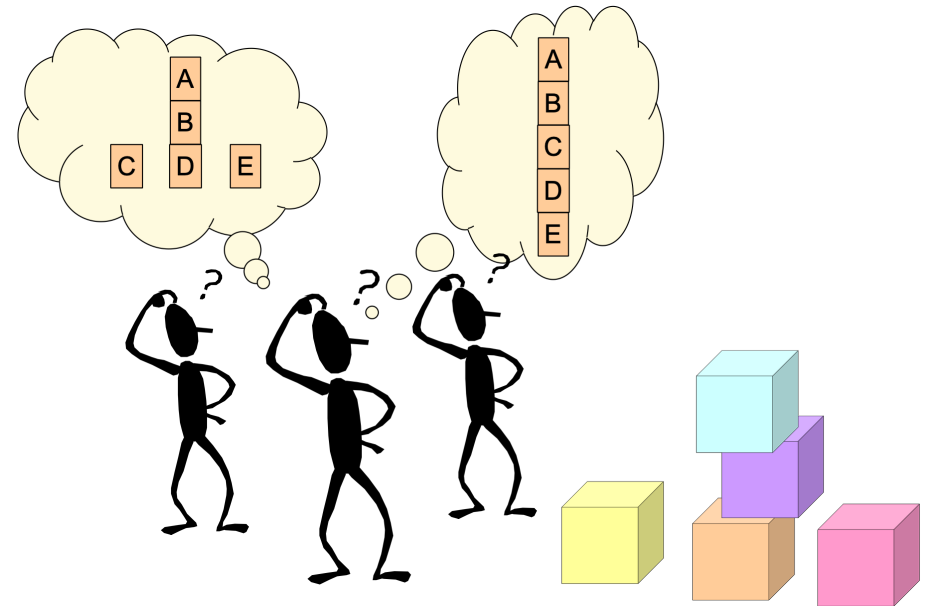
***Por lo menos significa la capacidad de comunicarse.***



- Cooperación es trabajar juntos como un equipo para conseguir un objetivo compartido.
- A menudo se requiere porque ningún agente puede conseguir el objetivo solo, o porque cooperando se obtendrá un resultado mejor (por ej., obtener un resultado más rápidamente).



- **Coordinación** es gestionar las interdependencias entre actividades.
- Por ejemplo si hay un recurso no compartible que se desea utilizar por más de una entidad (agente).

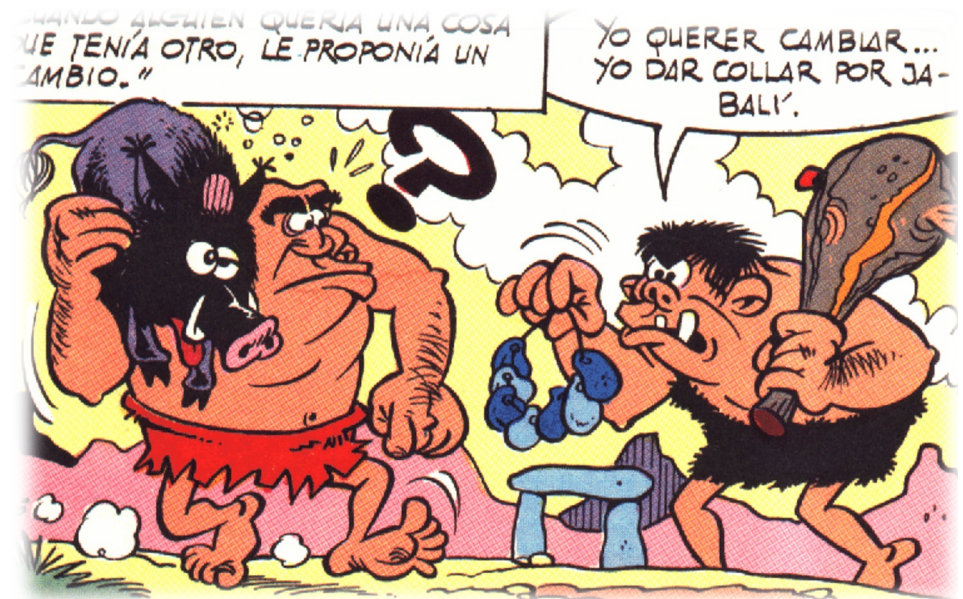




- La negociación es la capacidad de alcanzar acuerdos sobre temas de interés común.
- Por ejemplo: Estamos en casa delante del televisor, tu quieres ver un partido de futbol y tu pareja una película.

Un posible trato es ver el partido esta noche y mañana una película (este acuerdo puede tener un 'precio' que una parte ha de 'pagar' a la otra).

- Normalmente la negociación implica ofertas y contraofertas, con compromisos ('pagos') asumidos por los participantes.



## Concepto débil de agente

- Autonomía
- Proactividad
- Reactividad
- Sociabilidad

## Concepto fuerte de agente

Concepto débil +

- Movilidad
- Veracidad
- Benevolencia
- Racionalidad
- Aprendizaje / adaptación

Un agente **siempre** tiene las propiedades débiles de agencia y **puede** tener las propiedades fuertes





**Movilidad:** habilidad de trasladarse en una red de comunicación informática.

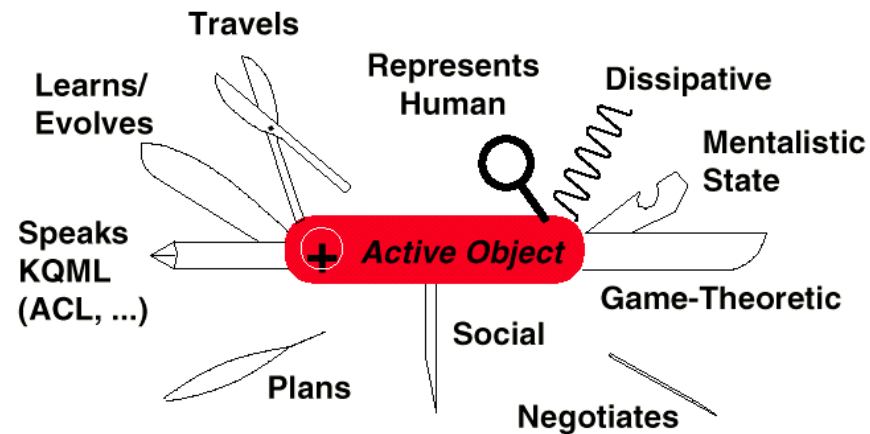
**Veracidad:** no comunica información falsa intencionadamente.

**Benevolencia:** no tiene objetivos contradictorios y siempre intenta realizar la tarea que se le solicita.

**Racionalidad:** tiene unos objetivos específicos y siempre intenta llevarlos a cabo.



Van Parunak:



(Also: Heterogeneous, Voting, Embodied...)



- En entornos complejos:
  - Un agente no tiene control completo sobre su entorno, sólo tiene un control parcial.
  - Control parcial significa que el agente puede influir sobre el entorno con sus acciones.
  - Una acción ejecutada por un agente puede fallar o tener el efecto deseado
- Conclusión: los entornos son no deterministas y los agentes deben estar preparados para posibles fallos.
- ¿Qué propiedades tienen los entornos y como influyen las mismas en los agentes?



- Accesible vs inaccesible.

Un entorno accesible es aquel en el que el agente puede obtener información completa, exacta y actualizada del estado del entorno.

Entornos moderadamente complejos (incluyendo por ejemplo, el mundo físico diario e internet) son inaccesibles.

Cuando más accesible es un entorno, más fácil es construir agentes que interactúen con él.



- **Determinista vs no determinista.**

Un entorno determinista es aquel en el que cualquier acción tiene un único efecto garantizado, no hay incertidumbre sobre el estado resultante de la ejecución de una acción.

El mundo físico puede a todos los efectos ser considerado como no determinista.

Los entornos no deterministas presentan grandes problemas para el diseñador de agentes.



- Episódico vs no episódico

En un entorno episódico el desempeño/actuación de un agente depende de un número discreto de episodios, no existiendo enlaces (relación) entre el desempeño de un agente en escenarios distintos.

Los entornos episódicos son, desde el punto de vista del desarrollador de agentes, más sencillos porque el agente puede decidir que acción ejecutar basándose únicamente en el episodio actual, no necesita razonar sobre las interacciones entre el episodio actual y los episodios futuros.





- Estático vs dinámico

Un entorno estático es aquel en el que se puede asumir que no se producen cambios excepto los provocados por la ejecución de acciones del agente.

Un entorno dinámico es aquel que tiene otros procesos que operan en el, y que por lo tanto se producen cambios que están fuera del control del agente.

El mundo físico es un entorno altamente dinámico.



- Discreto vs continuo

Un entorno es discreto si en él hay un número fijo y finito de acciones y percepciones.

El juego del ajedrez es un ejemplo de entorno discreto, y la conducción de un taxi un ejemplo de entorno continuo (AIMA, Russell and Norvig).



- Al explicar la actividad humana utilizamos declaraciones como la siguiente:

*Lola cogió su paraguas porque creía que estaba lloviendo y no quería mojarse.*

- Estas declaraciones hacen uso de una psicología popular, porque el comportamiento humano se predice y explica atribuyendo actitudes como: creer, desear, esperar, temer, ...
- Daniel Dennett acuñó el término **sistema intencional** al describir entidades "*cuyo comportamiento puede ser predicho por el método de atribuir creencias, deseos y perspicacia racional*".
- Un sistema intencional de primer orden tiene creencias y deseos (etc.), pero no creencias y deseos sobre creencias y deseos.
- Un sistema intencional de segundo orden es más sofisticado, tiene creencias y deseos (y sin duda otros estados intencionales) sobre creencias y deseos (y otros estados intencionales) - tanto los de los otros como de sí mismo.



- ¿Podemos utilizar la actitud intencional en máquinas?

Atribuir creencias, albedrio, intenciones, conciencia, habilidades, o deseos a una máquina es correcto cuando tal atribución expresa la misma información sobre la máquina que expresa sobre una persona.

Las teorías de la creencia, el conocimiento y el deseo se pueden construir para las máquinas de una forma más sencilla que para los seres humanos, y posteriormente aplicarse a los seres humanos.

Atribuir cualidades mentales es más sencillo para máquinas de estructura conocida, tales como termostatos y sistemas operativos, pero es más útil cuando se aplica a entidades cuya estructura no es completamente conocida (John McCarty)



- ¿Qué podemos describir con la actitud intencional ?

Consideremos un interruptor eléctrico:

*Atribuir creencias, albedrio, intenciones, conciencia, habilidades, o deseos a una máquina es correcto cuando tal atribución expresa la misma información sobre la máquina que expresa sobre una persona.*

Accionar el interruptor es simplemente nuestra forma de comunicar nuestros deseos (Yoav Shoham)



- La descripción de la actitud intencional es consistente,  
..... no nos convence cualquier cosa, ya que esencialmente comprendemos el mecanismo suficientemente como para tener una descripción más simple y **mecanicista** (*explica los fenómenos de la naturaleza mediante leyes automáticas de causa y efecto*) de su comportamiento. (Yoav Shoham)
- Cuanto más sepamos sobre un sistema, menos necesitamos confiar en explicaciones anímicas e intencionales de su comportamiento.





- Pero con sistemas muy complejos, una explicación mecanicista de su comportamiento puede no ser factible.
- Los sistemas informáticos son **cada vez más complejos**, por ello **necesitamos abstracciones y metáforas** más potentes para explicar su funcionamiento - especificaciones de bajo nivel son impracticables.
- La **actitud intencional** es una de tales abstracciones.



- Los **conceptos intencionales** son, pues, **herramientas de abstracción** que nos proporciona una forma cómoda y familiar de describir, explicar y predecir el comportamiento de los sistemas complejos.
- Hagamos memoria: los desarrollos más importantes en la informática se basan en nuevas abstracciones:
  - ✓ Abstracción procedural;
  - ✓ Tipos de datos abstractos;
  - ✓ Objetos.
- Los **agentes y los agentes como sistemas intencionales**, representan una abstracción más, y cada vez más, potente.



- Fortalezas de esta idea:

## Caracterización de Agentes

Nos proporciona una forma familiar, no técnica, de *comprender y explicar* agentes.

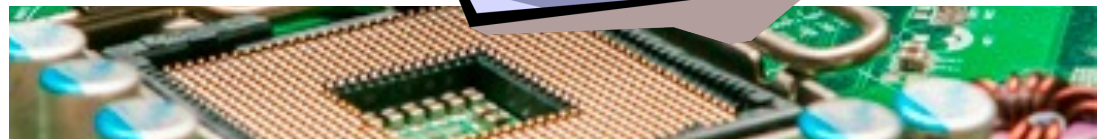
## Representaciones anidadas

Nos da la posibilidad de especificar sistemas que *incluyen representaciones de otros sistemas*.

Se acepta ampliamente que tales representaciones anidadas son esenciales para agentes que deben cooperar con otros agentes.



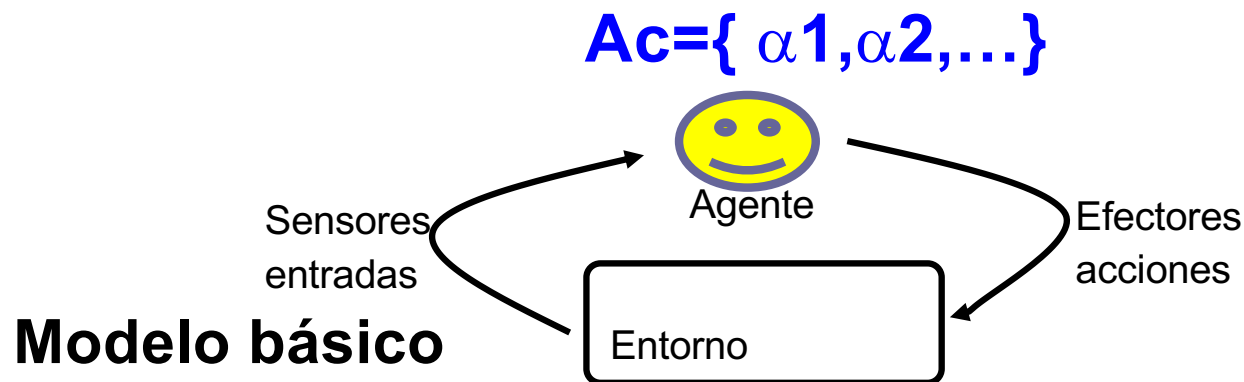
- Agentes
  - Agentes estándar
  - Agentes reactivos simples
  - Agentes con estado
- Entornos
- Història
- Percepción



- Asumimos que el entorno puede estar en uno cualquiera de los estados de un conjunto finito de estados instantáneos discreto (E):

$$E = \{ s_1, s_2, \dots \}$$

- Se asume que los agentes tienen disponible un repertorio (conjunto finito) de posibles acciones que transforman el estado del entorno :



- Un agente reactivo decide qué acción ejecutar sin tener en cuenta su historia (ninguna referencia al pasado) considerando solamente el presente.
- Formalmente el comportamiento de un agente reactivo se representa mediante una función:

**acción:  $E \rightarrow A$**

- ***Para cualquier agente reactivo hay un modelo de agente como el definido previamente, lo inverso no es generalmente cierto.***
- Un termostato es un agente reactivo.

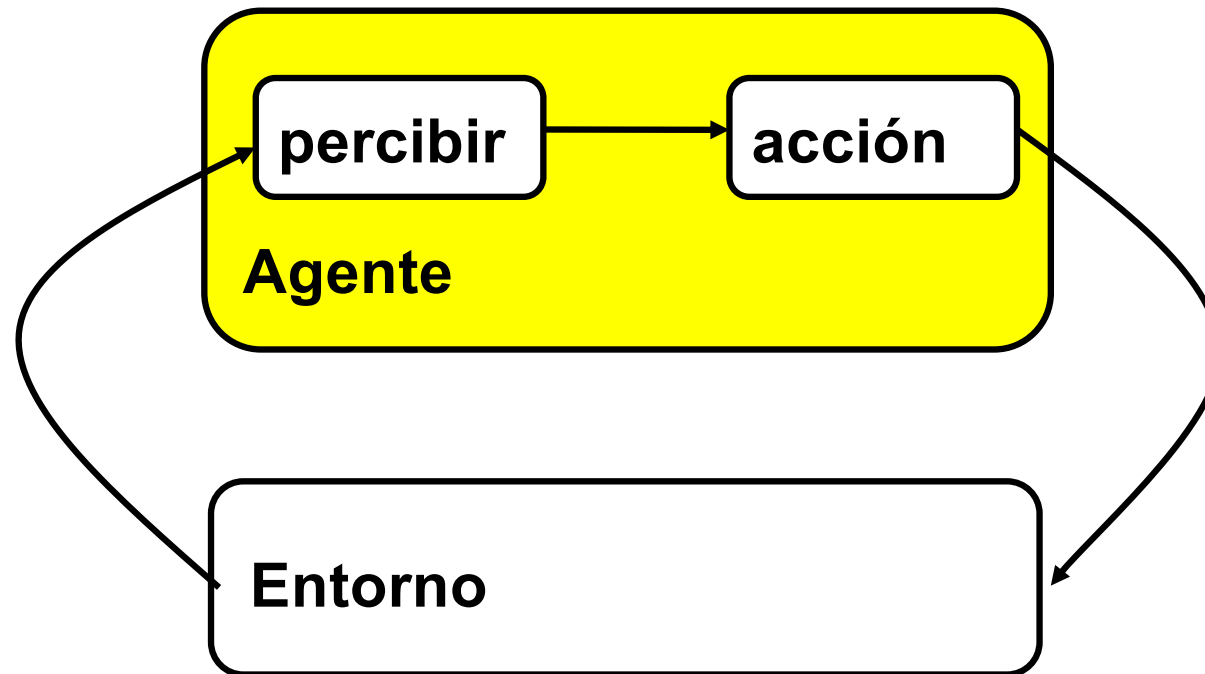
Estados del entorno = {temperatura OK, demasiado frio}

$$\text{acción(s)} = \begin{cases} \text{calentador off} & \text{si } s = \text{temperatura OK} \\ \text{calentador on} & \text{en cualquier otro caso} \end{cases}$$





- **Bucle de funcionamiento: Funciones *percibir* y *actuar*:**



- **La función *percibir* modela la capacidad del agente para percibir su entorno, mientras que la función *actuar* modela el proceso de toma de decisión del agente.**
- **La *Percepción* es el resultado de la función**  
***percibir: E → P***  
donde ***P*** es un conjunto (no vacío) de ***percepciones*** (entradas perceptivas), que relaciona estados del entorno con percepciones.
- La decisión del agente es el resultado de la función ***acción:***  
***acción: P\* → A***  
que relaciona secuencias de percepciones ( $P^*$ ) con acciones.



Ejemplo:

Si

$x$  = “La temperatura de la habitación es OK”

$y$  = “No hace calor en este instante”

entonces:

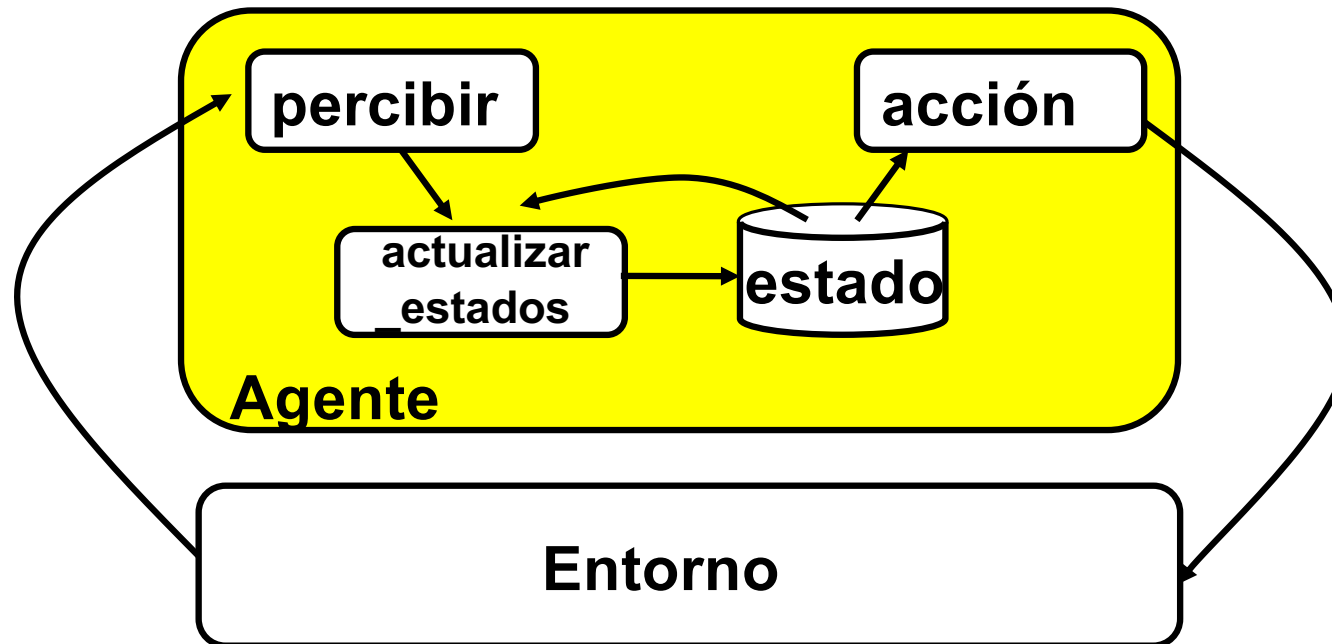
$$S = \{ \underset{s1}{(x, y)}, \underset{s2}{(x, \neg y)}, \underset{s3}{(\neg x, y)}, \underset{s4}{(\neg x, \neg y)} \}$$

pero para el termostato:

$$\text{percibir}(s) = \begin{cases} p1 \text{ si } s=s1 \text{ o } s=s2 \\ p2 \text{ si } s=s3 \text{ o } s=s4 \end{cases}$$



- Ahora consideramos agentes que mantienen el estado, su bucle de funcionamiento es:***



**Puntualización:** considerar agentes a este nivel de abstracción facilita su análisis, pero no nos ayuda a construirlos.



- Estos agentes tienen una estructura de datos interna que es utilizada para recordar información sobre la historia y estado del entorno.
- Tienen la misma función de percepción:

$$\textit{percibir}: E \rightarrow P$$

- La función acción-selección es definida ahora como mapping entre estados internos y acciones:

$$\textit{acción}: I \rightarrow Ac$$

donde:

**$I$ : es el conjunto de todos los estados internos del agente**

- Se introduce una nueva función de siguiente estado que define un estado interno a partir de un estado interno y las percepciones :

$$\textit{actualizar\_estados}: I \times P \rightarrow I$$

Esta función determinara el nuevo estado interno a partir del estado interno en que se encuentre el agente y las percepciones que perciba en su entorno.

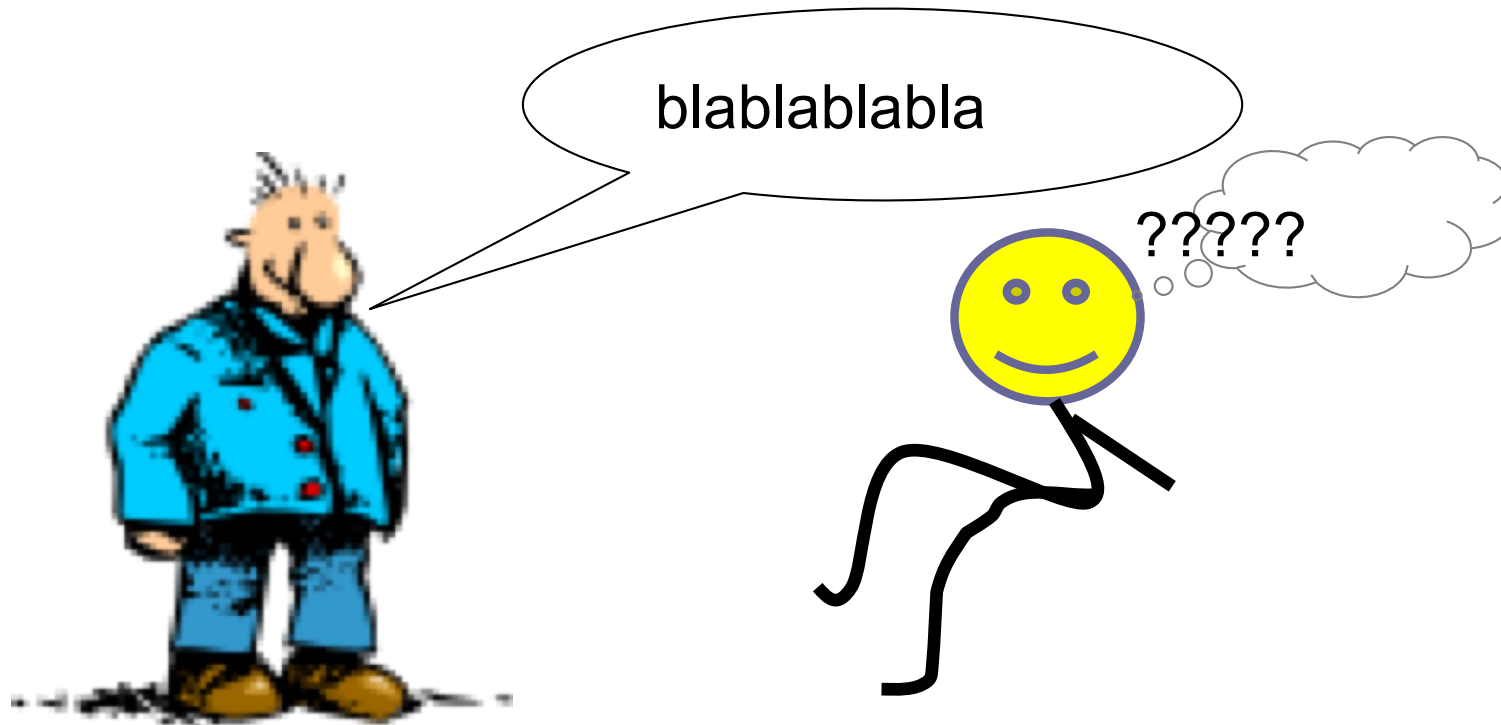


1. El agente inicia su funcionamiento e algún estado inicial  $i_0$ ,  
 $i_0 = (\text{actualizar\_estados}(s_0, \text{percibir}(s_0))); i = 1;$
2. Repetir (repeat forever)
  - 2.1 Observar el estado del entorno  $s_i$  y generar una percepción mediante la función de percepción  
 $p_i = \text{percibir}(s_i)$
  - 2.2 Actualizar el estado interno mediante la función siguiente estado  
 $i_i = \text{actualizar\_estados}(i_{i-1}, p_i)$
  - 2.3 La acción seleccionada por el agente será:  
 $a_i = \text{acción}(i_i)$
  - 2.4 Ejecutar la acción  $a_i$ ,  $i = i + 1$





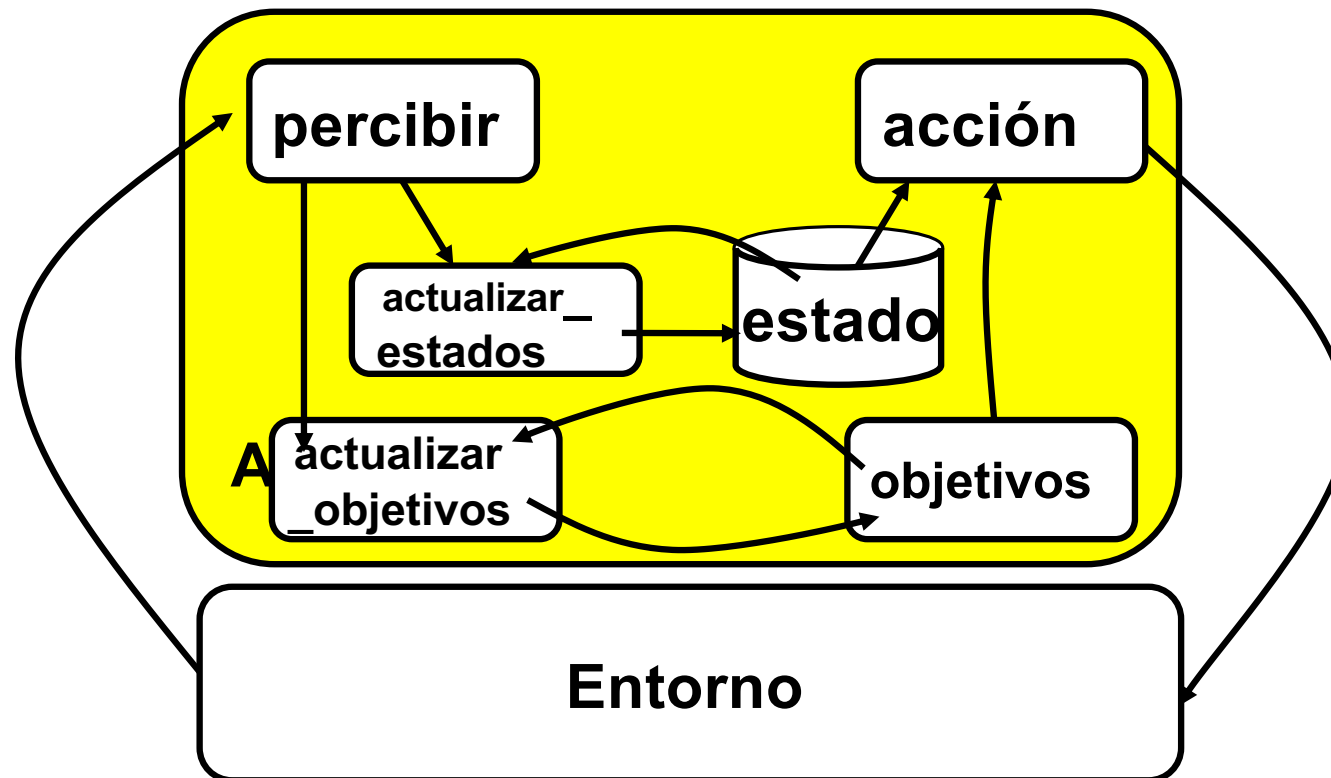
# **etsinf** ¿Cómo le decimos a un agente lo que tiene que hacer?



Mediante objetivos  
Mediante funciones de utilidad



- Ahora consideramos agentes que mantienen el estado y que tienen objetivos que satisfacer, su bucle de funcionamiento es:*



Estos agentes tienen la misma estructura y funciones que los agentes con estado (percibir, siguiente), se introduce un nuevo conjunto, el conjunto de objetivos  $G$  (cada uno de ellos es una descripción de un estado o conjunto de estados meta del agente –estados del entorno que el agente desea alcanzar-).

Se introduce una nueva función para actualizar los objetivos del agente a partir de las percepciones del mismo:

$$\text{actualizar\_objetivos: } G \times P \rightarrow G$$

- ***Y redefinimos la función acción:***

$$\text{acción: } I \times G \rightarrow Ac$$

***La acción seleccionada por el agente dependerá del estado interno y de los objetivos que el agente quiera alcanzar.***



1. El agente inicia su funcionamiento en algún estado inicial  $s_0$ ,  
 $l_0 = \text{actualizar\_estados}(s_0, \text{percibir}(s_0)); i = 1;$
2. Repetir (repeat forever)
  - 2.1 Observar el estado del entorno  $s_i$  y generar una percepción mediante la función de percepción  
 $p_i = \text{percibir}(s_i)$
  - 2.2 Actualizar el estado interno mediante la función siguiente estado  
 $l_i = \text{actualizar\_estados}(l_{i-1}, p_i)$
  - 2.3 Actualizar los objetivos a partir de las nuevas percepciones  
 $g = \text{actualizar\_objetivos}(l_i, g)$
  - 2.4 La acción seleccionada por el agente será:  
 $a_i = \text{acción}(l_i, g)$
  - 2.5 Ejecutar la acción  $a_i$ ,  $i = i + 1$

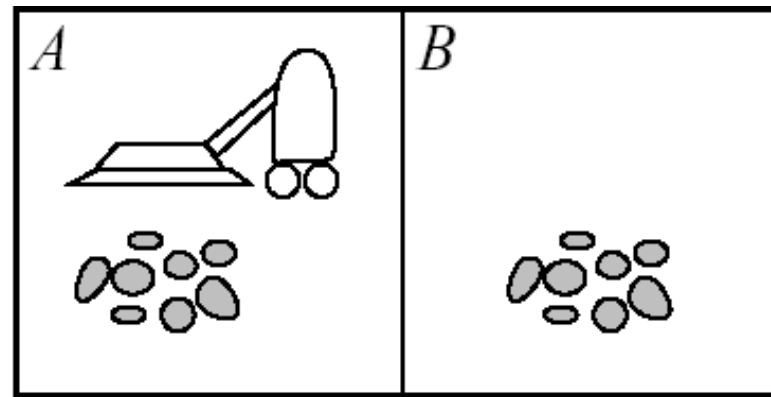


Construimos agentes para que realicen tareas por nosotros

Las tareas deben ser especificadas por nosotros ....

Queremos decirle a los agentes qué tienen que hacer, pero sin decirle cómo lo tienen que hacer.





Percepciones: localización y contenidos, por ejemplo,  
 $[A, \text{Sucio}]$ .

Acciones: *Izquierda, Derecha, Aspirar, NoOp*.





Secuencia de percepciones	Acción
$[A, Limpio]$	<i>Derecha</i>
$[A, Sucio]$	<i>Aspirar</i>
$[B, Limpio]$	<i>Izquierda</i>
$[B, Sucio]$	<i>Aspirar</i>
$[A, Limpio], [A, Limpio]$	<i>Derecha</i>
$[A, Limpio], [A, Sucio]$	<i>Aspirar</i>
$[A, Clean], [A, Clean]$	<i>Right</i>
$[A, Clean], [A, Dirty]$	<i>Suck</i>
$\vdots$	$\vdots$

**función** AGENTE-ASPIRADORA-REACTIVO([localización, estado]) **devuelve** una acción

**Si** estado = *Sucio* **entonces** {devolver *Aspirar*}  
**sino** {**si** localización = A **entonces** {devolver *Derecha*}  
           **sino** {**si** localización = B **entonces** {devolver *Izquierda*}}}



Una única **medida de rendimiento** evalúa la secuencia del entorno:

- ❖ ¿Un punto por cada recuadro eliminado en un tiempo  $T$ ?
- ❖ ¿Un punto por cada recuadro limpio por período, menos un punto por movimiento?
- ❖ ¿Con penalización para  $> k$  recuadros sucios?

Un **agente racional** emprende aquella acción que maximice el valor **esperado** de la medida de rendimiento, **basándose en las evidencias aportadas por la secuencia de percepciones**.

Racional  $\neq$  omnisciente.

Racional  $\neq$  clarividente.

Racional  $\neq$  exitoso.



Para diseñar un agente racional, debemos especificar el **entorno de trabajo**.

Por ejemplo, considere, el entorno de diseñar un taxi automático:

¿Cuál es la medida de rendimiento? Seguridad, destino, beneficios, legal, viaje confortable ...

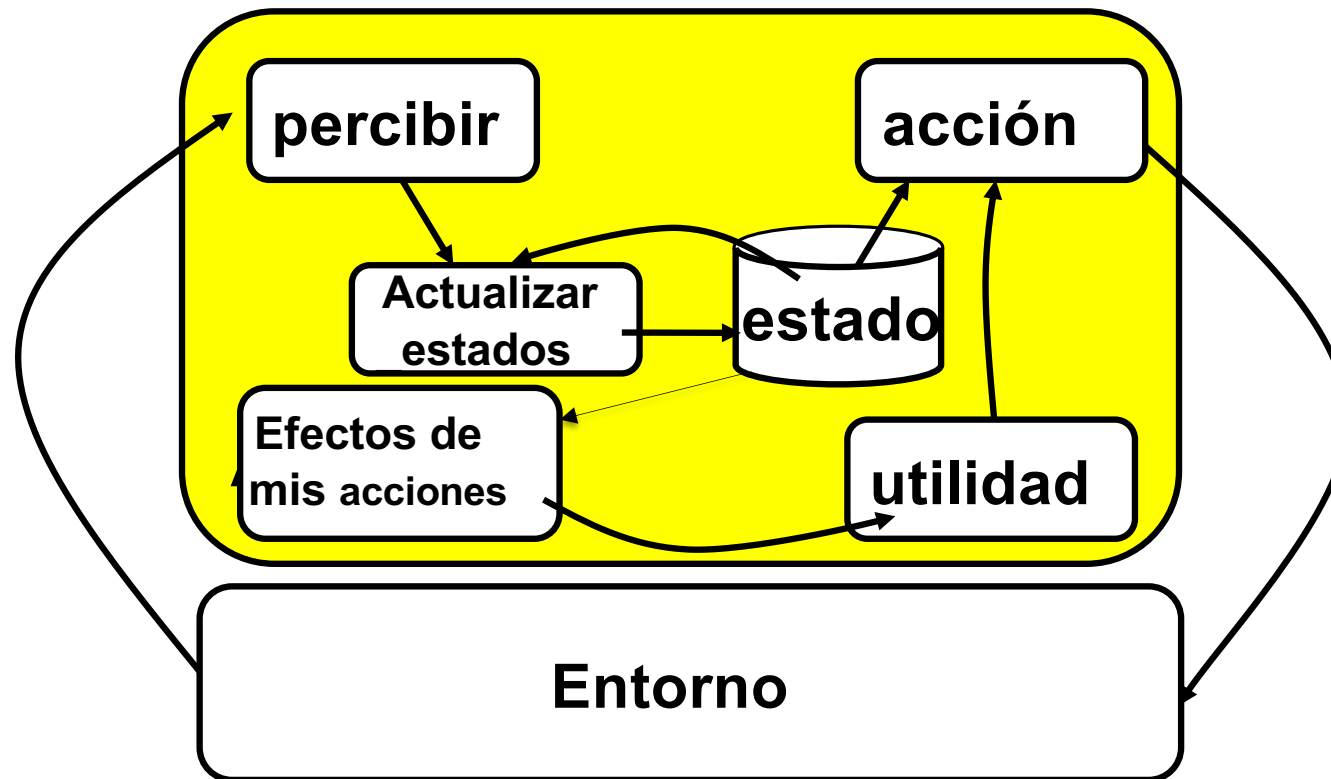
¿Cuál es el entorno? Carreteras, tráfico, peatones, el tiempo ...

¿Cuáles son los actuadores? Dirección, acelerador, freno, bocina, visualizador ...

¿Cuáles son los sensores? Cámara, visualizador de la aceleración, indicador de la gasolina, sensores del motor, teclado, GPS ...



El bucle de funcionamiento es:



Una utilidad es un valor numérico que representa cuan 'bueno' es un estado, cuando mayor es la utilidad, mejor es el estado.

La tarea del agente es alcanzar estados (seleccionar acciones) que maximicen la utilidad (al agente no se le indica que tiene que hacer para alcanzar los estados).

Una especificación de tarea es una función

$$u: E \rightarrow \mathbb{R}$$

que asocia un número real a cada estado del entorno.



¿Pero cual es el valor de una **ejecución** de agente?

- ¿La mínima utilidad del estado de una ejecución? (pesimista)
- ¿La máxima utilidad del estado de una ejecución? (optimista)
- ¿La suma de las utilidades de los estados de una ejecución?
- ¿La media?

Desventajas: al asignar utilidades a estados locales, es difícil  
precisar una visión a largo plazo.





Una solución: no asignar una utilidad a estados individuales sino a las ejecuciones del agente:

$$u: \mathcal{R} \rightarrow \mathbb{R}$$


- Esta aproximación tiene inherentemente una visión a largo plazo.
- Otras variaciones, incorporar las probabilidades de los diferentes estados emergentes.
- En cada decisión del agente de qué acción elegir, elegir aquella que da lugar al estado de mayor utilidad.



- ¿De donde vienen los números? (Peter Cheeseman)
- La gente no piensa en términos de utilidades, es difícil para las personas especificar las tareas en estos términos.
- Sin embargo, funciona bien en ciertos escenarios. . .



- Simulación de un entorno de una cuadrícula dimensional sobre el que hay agentes, baldosas, obstáculos y agujeros.
- Un agente puede moverse en cuatro direcciones: arriba, abajo, izquierda o derecha, y si está localizado al lado de una baldosa puede empujarla.
- Los agujeros tienen que ser rellenados por el agente con las baldosas. Un agente suma puntos rellenando huecos con baldosas, su objetivo es llenar tantos agujeros como sea posible.
- El 'mundo de las baldosas' cambia con la aparición y desaparición aleatoria de agujeros.

	agujero	agujero		
		baldosa		
			baldosa	

		agujero		
		baldosa		
			baldosa	agujero
agujero				



- Definimos la función de utilidad con la siguiente ecuación:

$$u(r) = \frac{\text{número de agujeros rellenados en } r}{\text{número total de agujeros que hay en } r}$$

- Entonces:
  - Si el agente rellena todos los agujeros, utilidad = 1.
  - Si el agente no rellena ningún agujero, utilidad = 0



- Denotamos por  $P(r|Ag, Env)$  la probabilidad de que ocurra la ejecución  $r$  cuando el agente  $Ag$  está situado en el entorno  $Env$ . Observar que se tiene que cumplir que:

$$\sum_{r \in R(Ag, Env)} P(r | Ag, Env) = 1$$

- La utilidad esperada del agente  $Ag$  en el entorno  $Env$  (dados  $P, u$ ) es:

$$EU(Ag, Env) = \sum_{r \in R(Ag, Env)} U(r) P(r | Ag, Env) \quad (1)$$



- Consideremos el entorno  $Env1 = \langle E, s_0, \tau \rangle$  definido de la siguiente forma:

$$E = \{s_0, s_1, s_2, s_3, s_4, s_5\}$$

$$\tau(s_0 \xrightarrow{\alpha_1}) = \{e_3, e_4, e_5\}$$

$$\tau(s_0 \xrightarrow{\alpha_0}) = \{e_1, e_2\}$$

- Hay dos posibles agente con respecto a este entorno:

$$Ag_1(e_0) = \alpha_0$$

$$Ag_2(e_0) = \alpha_1$$





Las probabilidades de las diversas ejecuciones son:

$$P(e_0 \xrightarrow{\alpha_0} e_1 | Ag_1, Env_1) = 0.6$$

$$P(e_0 \xrightarrow{\alpha_1} e_2 | Ag_1, Env_1) = 0.4$$

$$P(e_0 \xrightarrow{\alpha_1} e_3 | Ag_2, Env_1) = 0.1$$

$$P(e_0 \xrightarrow{\alpha_1} e_4 | Ag_2, Env_1) = 0.2$$

$$P(e_0 \xrightarrow{\alpha_1} e_5 | Ag_2, Env_1) = 0.7$$

Asumiendo que la función de utilidad es (1):

$$u(e_0 \xrightarrow{\alpha_0} e_1) = 8$$

$$u(e_0 \xrightarrow{\alpha_1} e_2) = 11$$

$$u(e_0 \xrightarrow{\alpha_1} e_3) = 70$$

$$u(e_0 \xrightarrow{\alpha_1} e_4) = 9$$

$$u(e_0 \xrightarrow{\alpha_1} e_5) = 10$$

¿Cuáles son las utilidades esperadas de estos agentes para esta función de utilidad?



¿Cuáles son las utilidades esperadas de estos agentes para esta función de utilidad?

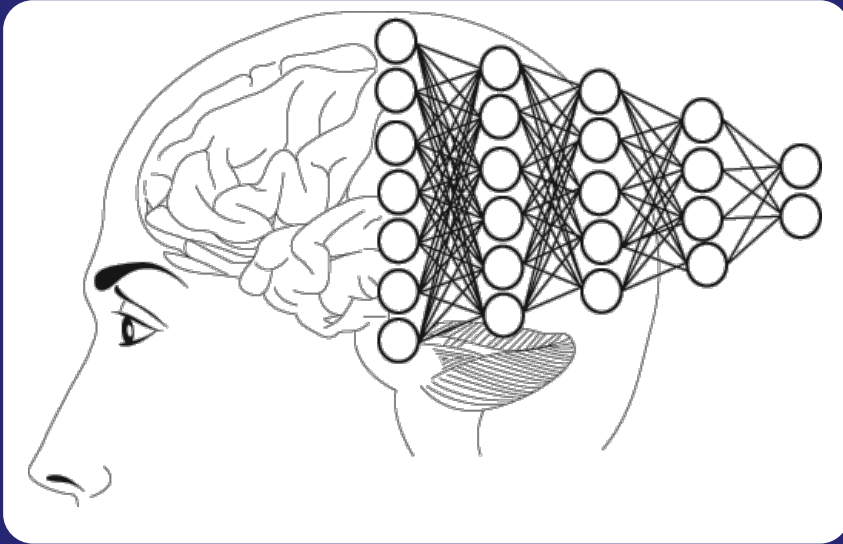
$$EU(Ag_i, Env_1) = \sum_{r \in R(Ag, Env)} U(r)P(r | Ag_i, Env_1)$$

$$\begin{aligned} EU(Ag_1, Env_1) &= u(e_0 \xrightarrow{\alpha_0} e_1) \times P(e_0 \xrightarrow{\alpha_0} e_1 | Ag_1, Env_1) + u(e_0 \xrightarrow{\alpha_0} e_2) \times P(e_0 \xrightarrow{\alpha_0} e_2 | Ag_1, Env_1) = \\ &= 8 \times 0.6 + 11 \times 0.40 = 9,2 \end{aligned}$$

$$\begin{aligned} EU(Ag_2, Env_1) &= u(e_0 \xrightarrow{\alpha_1} e_3) \times P(e_0 \xrightarrow{\alpha_1} e_3 | Ag_2, Env_1) + u(e_0 \xrightarrow{\alpha_1} e_4) \times P(e_0 \xrightarrow{\alpha_1} e_4 | Ag_2, Env_1) + \\ &u(e_0 \xrightarrow{\alpha_1} e_5) \times P(e_0 \xrightarrow{\alpha_1} e_5 | Ag_2, Env_1) = 70 \times 0.1 + 9 \times 0.20 + 10 \times 0.7 = 15.8 \end{aligned}$$



# Razonamiento Práctico

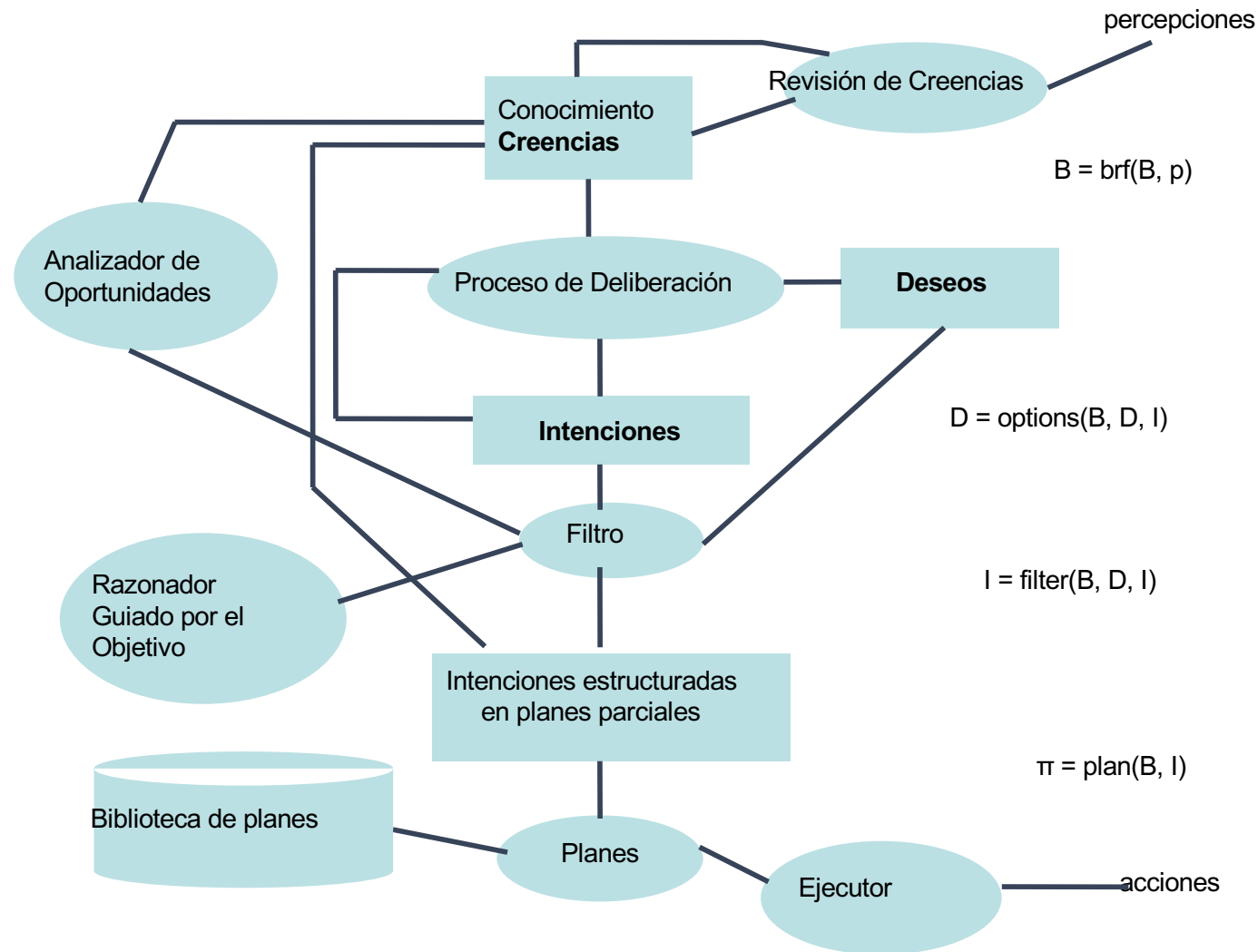


Razonamiento práctico es razonamiento dirigido *por las acciones* - el proceso de descubrir qué hacer:

- *El razonamiento práctico es una cuestión de sopesar consideraciones conflictivas por y contra las opciones que compiten, donde las consideraciones relevantes se proveen por lo que el agente desea/valora/le importa y lo que el agente cree. (Bratman)*

El Razonamiento Práctico se distingue del Razonamiento Teórico en que el *razonamiento teórico* es dirigido por las *creencias*.





## Bucle de control del agente

*while true*

- ¿cuáles son las opciones (deseos)?
- ¿cómo elegir una opción?
- incl. filtrado
- opción elegida: intención ...

*observe the world;*

*update internal world model;*

*deliberate about what intention to achieve next;*

*use means-ends reasoning to get a plan for the*  
*intention;*

*execute the plan*

*end while*

¿cuándo reconsiderar las intenciones?



Hagamos el algoritmo más formal:

```
Agent Control Loop Version 2
1.   $B := B_0$ ; /* initial beliefs */
2.  while true do
3.      get next percept  $\rho$ ;
4.       $B := brf(B, \rho)$ ;
5.       $I := deliberate(B)$ ;
6.       $\pi := plan(B, I)$ ;
7.      execute( $\pi$ )
8.  end while
```







## Agent Control Loop Version 7

```
1.
2.   $B := B_0;$ 
3.   $I := I_0;$ 
4.  while true do
5.      get next percept  $\rho$ ;
6.       $B := brf(B, \rho);$ 
7.       $D := options(B, I);$ 
8.       $I := filter(B, D, I);$ 
9.       $\pi := plan(B, I);$ 
10.     while not ( $empty(\pi)$ 
11.                or  $succeeded(I, B)$ 
12.                or  $impossible(I, B)$ ) do
13.          $\alpha := hd(\pi);$ 
14.          $execute(\alpha);$ 
15.          $\pi := tail(\pi);$ 
16.         get next percept  $\rho$ ;
17.          $B := brf(B, \rho);$ 
18.         if  $reconsider(I, B)$  then
19.              $D := options(B, I);$ 
20.              $I := filter(B, D, I);$ 
21.         end-if
22.         if not  $sound(\pi, I, B)$  then
23.              $\pi := plan(B, I)$ 
24.         end-if
25.     end-while
26. end-while
```

*Control meta-nivel*

