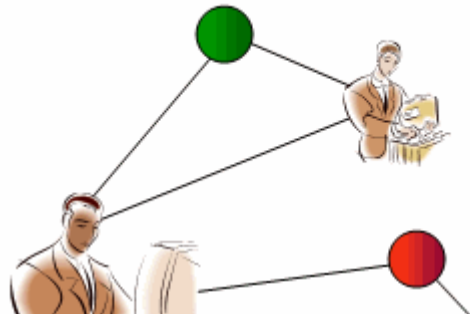
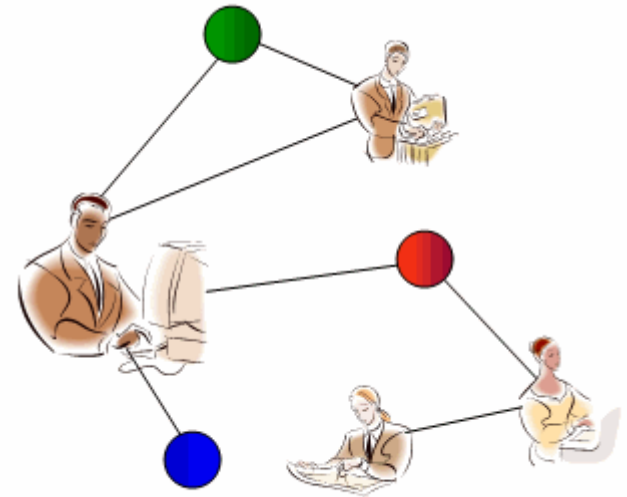


INTRODUCCION AGENTES Y SISTEMAS MULTIAGENTES

Reseña

- Cinco tendencias que han marcado hitos en la historia de la computación:
 - ❑ *ubicuidad*;
 - ❑ *interconexión*;
 - ❑ *inteligencia*;
 - ❑ *delegación*; y
 - ❑ *Orientación hacia lo humano*
- 
- The diagram illustrates the five trends in computing history. It features a network of three nodes: a green node at the top, a red node at the bottom right, and a yellow node at the bottom left. Lines connect the green node to the red node, the green node to the yellow node, and the red node to the yellow node. Each node is associated with a person: the green node is connected to a person in a suit, the red node is connected to a person in a suit, and the yellow node is connected to a person in a suit. The person connected to the yellow node is holding a stack of papers.



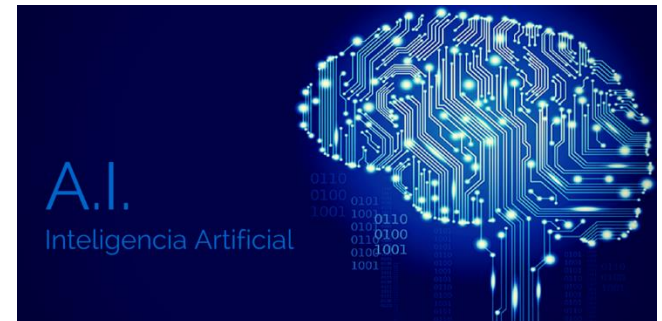
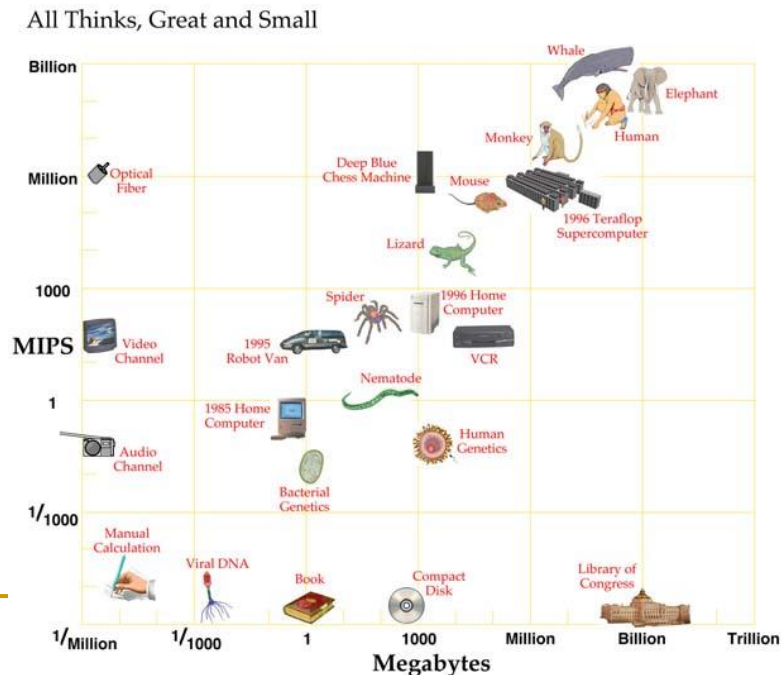
Interconexión

- Los SIC no son islas, ahora se conectan en redes y forman sistemas distribuidos
- La internet es un ejemplo obvio, pero las redes están alargando sus tentáculos ...
- Como los SIC distribuidos y concurrentes se han vuelto la norma, los investigadores han desarrollado modelos que ven la computación como un proceso de interacción



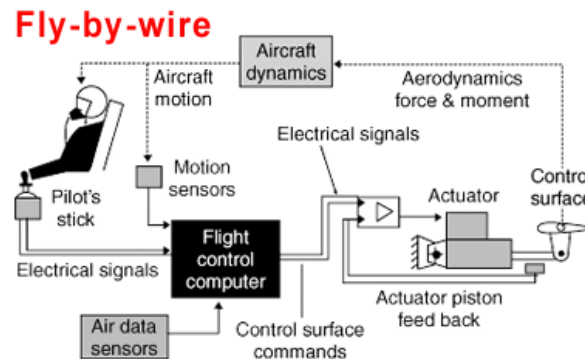
Inteligencia

- La complejidad de las tareas que podemos automatizar y delegar a los computadores han ido aumentando sostenidamente
- Si uno no se siente cómodo con eso, es probablemente debido a que uno es un ser humano



Delegación

- Los computadores están haciendo más cosas por nosotros – sin nuestra intervención
- Estamos *dándoles el control* a los computadores, incluso en tareas de seguridad críticas
- Un ejemplo: aviones “fly-by-wire”, donde se le cree más a un computador que a un piloto experimentado
- Otros: automóviles drive-by-wire, sistemas inteligentes de frenado, control de crucero que mantiene una distancia de seguridad con auto en el frente ...



Orientación hacia el humano

- Movimiento en programación alejándose de la orientación hacia la máquina y acercándose a metáforas que reflejen más la forma en que nosotros entendemos el mundo
- Los programadores (y usuarios) se relacionan en forma diferente con las máquinas
- Los programadores conceptualizan y construyen software en término de abstracciones de más alto nivel más cercanas a los humanos

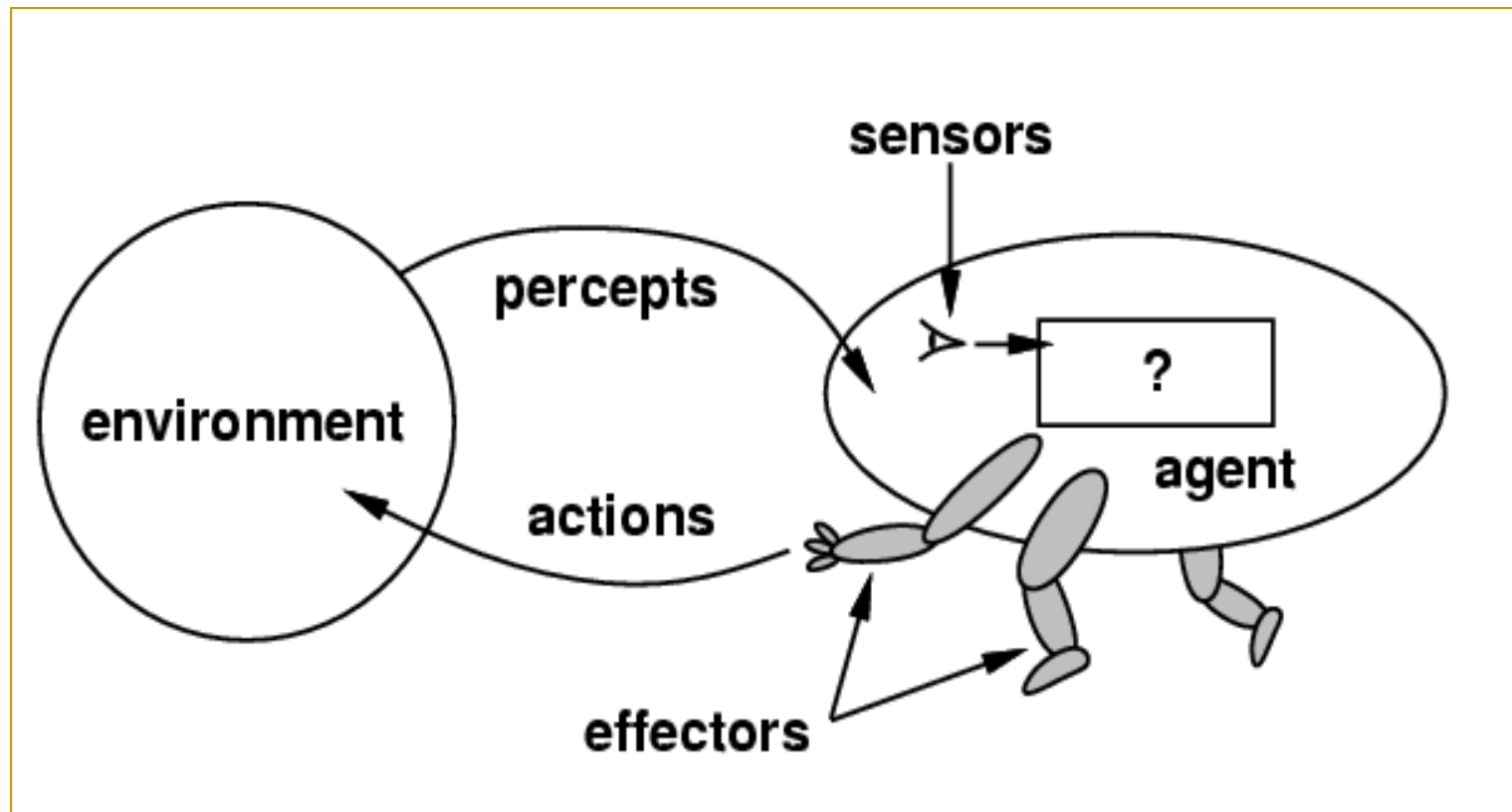


¿A dónde nos lleva todo eso?

- La Delegación e Inteligencia implican la necesidad de construir sistemas de computación que puedan actuar efectivamente a nuestro nombre
- Esto implica:
 - La habilidad de SC para actuar *independientemente*
 - La habilidad de SC para actuar de una forma que *represente nuestros mejores intereses* cuando interactuen con otros sistemas o humanos

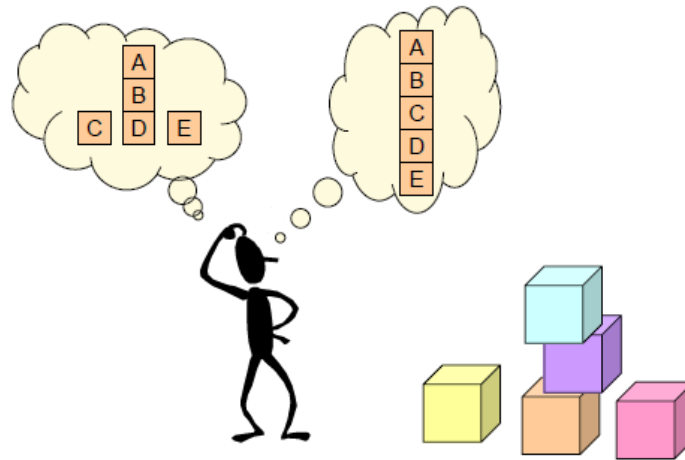


AGENTE



Agente, una definición

- Una agente es un sistema computacional que es capaz de actuar *independientemente* a nombre de un usuario o dueño, (determinando qué debe hacer para satisfacer un objetivo de diseño, en vez de tener que constantemente decirle que debe hacer)



AGENTE INTELIGENTE

Noción Débil:

Es la forma más general en que es usado el término agente. Es un sistema de software (hardware) con las siguientes propiedades:

- ↪ Autonomía
- ↪ Habilidad Social
- ↪ Reactividad
- ↪ Proactividad

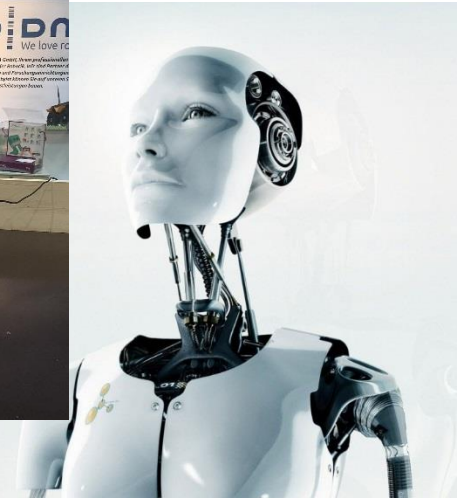


AGENTE INTELIGENTE

Noción más fuerte:

Además de las propiedades anteriores, se agregan *nociones mentales* tales como:

- Conocimiento
- Creencias
- Intenciones
- Obligaciones
- (Emociones)



Reactividad

- Si el ambiente donde corre un programa es inalterable, el programa no debe preocuparse por su éxito o fracaso – simplemente debe ejecutarse “a ciegas”
 - Ejemplo de un Sistema TPDV
- El mundo real es cambiante: las cosas cambian, la información es incompleta. Muchos (¿La mayoría?) de los ambientes interesantes son *dinámicos*
- Es difícil construir Software para dominios dinámicos: el programa debe tomar en cuenta la posibilidad de fallar – ¡ Incluso preguntarse si vale la pena ejecutarse!
- Un sistema *reactivo* mantiene una interacción a través del tiempo con su ambiente y responde a cambios en este (en un tiempo razonable para que la reacción sea útil)

Proactividad

- Reaccionar a un ambiente es fácil (estímulo → reglas de respuesta)
- Pero generalmente queremos agentes que *hagan cosas para nosotros*
- Es decir, *comportamiento orientado al objetivo*
- Pro-actividad = generar y tratar de alcanzar los objetivos; motivado no solo por los eventos; tomando la iniciativa
- Reconocer oportunidades



Balanceo de Reactividad y Comportamiento Orientado al Objetivo

- Queremos agentes reactivos, que respondan a condiciones cambiantes en forma apropiada y oportuna
- Queremos agentes que trabajen sistemáticamente para lograr objetivos de largo plazo
- Estas dos consideraciones pueden contraponerse
- Diseñar un agente que concilie esas consideraciones es aún un desafío



Habilidad Social

- El mundo real es un ambiente “multiagente”: no podemos intentar alcanzar nuestros objetivos sin considerar al resto
- Algunos objetivos solo se pueden lograr con la cooperación de otros
- Similarmente para varios ambientes computacionales : por ejemplo, la Internet
- *Habilidad Social* en un agente es la habilidad para interactuar con otros agentes (y posiblemente humanos) a través de algún tipo de *lenguaje de comunicación de agentes*, y tal vez cooperar con otros



Otras Propiedades

- Otras propiedades, a veces discutidas en el contexto de agentes:
- *movilidad*: la habilidad de un agente para moverse en una red de comunicaciones
- *veracidad*: un agente no comunicará información falsa a sabiendas
- *benevolencia*: los agentes no tienen objetivos conflictivos, por lo que un agente siempre intentará hacer lo que se le pide
- *racionalidad*: un agente actuará para lograr sus objetivos, y NO actuará a sabiendas para no alcanzarlos, al menos en el mejor de su conocimiento
- *aprendizaje/adaptación*: los agentes mejoran su rendimiento a través del tiempo

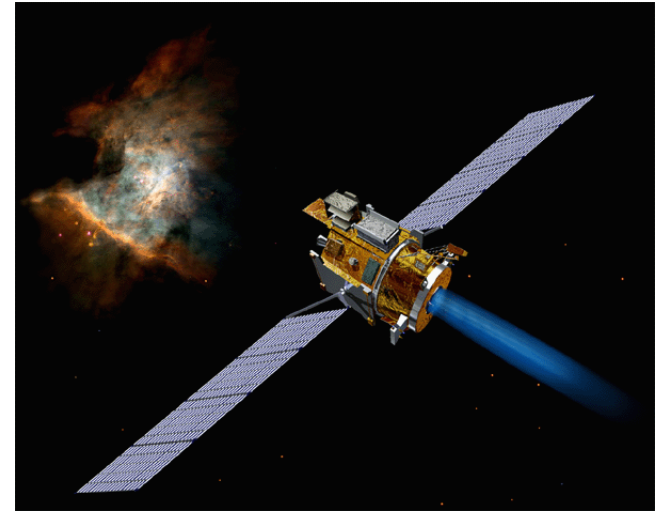
Control de naves espaciales

- Cuando una sonda espacial viaja desde la Tierra a los planetas, un grupo de control debe usualmente monitorear continuamente su viaje, y decidir cómo lidiar con problemas inesperados.
- Esto es costoso, y si se requieren decisiones *rápidas*, no es posible.
- Por estas razones, organizaciones como NASA están seriamente investigando la posibilidad de construir sondas más autónomas — dándoles capacidades de decisión más grandes y mayores responsabilidades.
- *Esto no es ficción: la sonda DS1 de NASA lo tiene*

Deep Space 1

- <http://nmp.jpl.nasa.gov/ds1/>

- “Deep Space 1 launched from Cape Canaveral on October 24, 1998. During a highly successful primary mission, it tested 12 advanced, high-risk technologies in space. In an extremely successful extended mission, it encountered comet Borrelly and returned the best images and other science data ever from a comet. During its fully successful hyperextended mission, it conducted further technology tests. The spacecraft was retired on December 18, 2001.”
– NASA Web site



Agentes Autónomos para tareas especializadas

- El ejemplo de la DS1 es uno genérico
- Agentes (y su instanciación física en robots) juegan un rol importante en situaciones de alto riesgo, no apropiadas o imposibles para ser enfrentadas por humanos
- El grado de autonomía variará dependiendo de la situación (usar control remoto humano puede ser una alternativa, pero no siempre)



Ambiente de la tarea

- Antes de diseñar un agente, debemos especificar su “ambiente de tarea”:

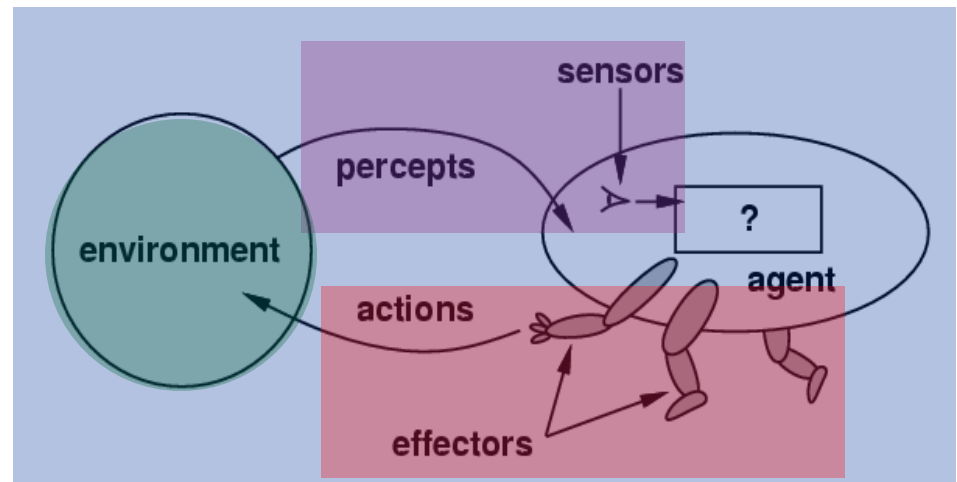
PEAS:

Performance measure

Environment

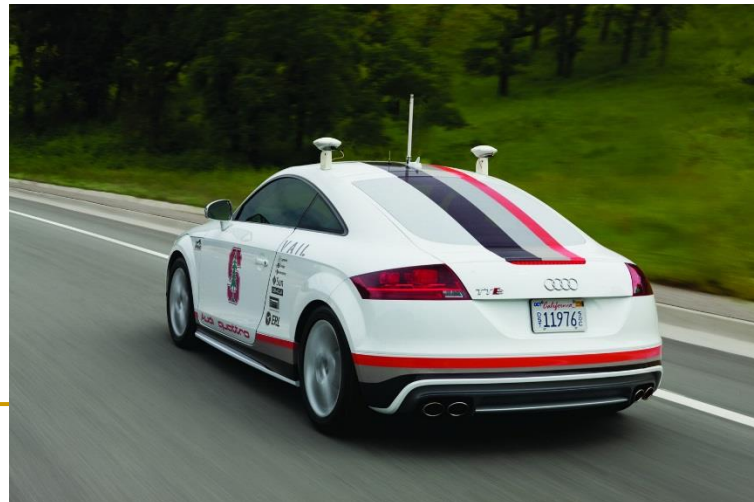
Actuators

Sensors



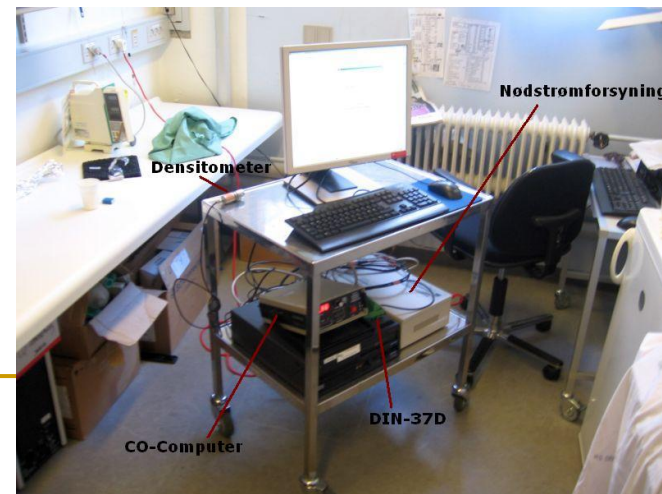
PEAS

- Ejemplo: Agente = conductor de taxi (vehículo conducción autónoma)
 - ❑ **Performance measure:** seguro, rápido, legal, viaje comfortable, maximizar utilidad
 - ❑ **Environment:** rutas, otro tráfico, peatones, clientes
 - ❑ **Actuators:** manubrio, acelerador, freno, señalética, bocina
 - ❑ **Sensors:** cámara, sonar, velocímetro, GPS, odómetro, sensores del motor, teclado



PEAS

- Ejemplo: Agente = Sistema de diagnóstico médico
- **Performance measure:** paciente sano, minimizar costos, demandas
- **Environment:** Paciente, hospital, personal
- **Actuators:** Pantalla de salida (preguntas, tests, diagnósticos, tratamientos, consultas)
- **Sensors:** Teclado (entrada de síntomas, conclusiones, respuestas del paciente)



PEAS

- Ejemplo: Agente = robot que traslada y almacena partes
- **Performance measure:** Porcentaje de partes guardadas en el contenedor correcto
- **Environment:** correa transportadora con las partes, contenedores
- **Actuators:** brazo y mano articulados
- **Sensors:** Cámara, sensores de ángulo en juntas brazo, mano



Ejercicio práctico: PEAS

- Ejemplo: Agente = robot aspiradora (Roomba y Braava © iRobot)
- Performance measure:
- Environment:
- Actuators:
- Sensors:



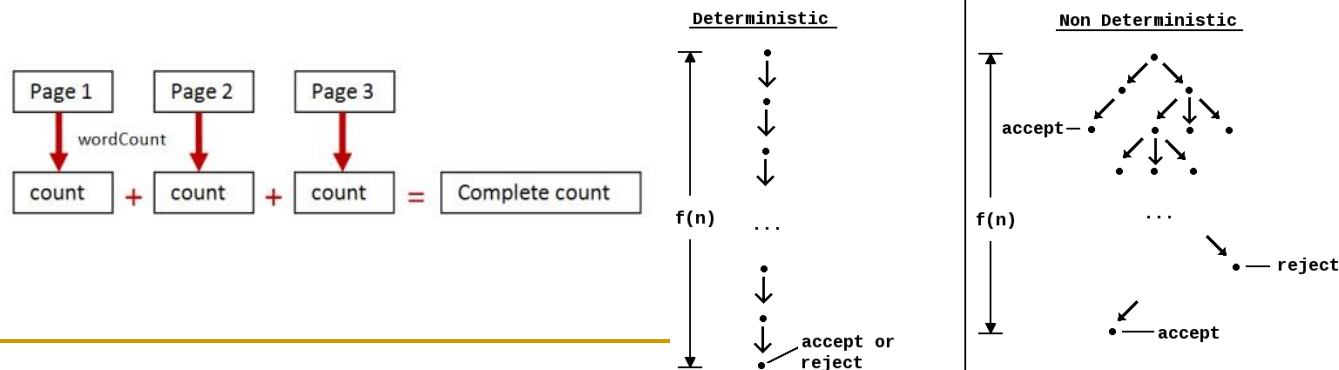
Ambientes – *Accesibles vs. inaccesibles*

- Un ambiente accesible es aquel en que un agente puede obtener información completa, precisa y actualizada del estado de dicho ambiente
- La mayoría de los ambientes moderadamente complejos (incluyendo, p. ej., el mundo físico e Internet) son inaccesibles
- Mientras más accesible un ambiente, más fácil es construir agentes que operen bien en él



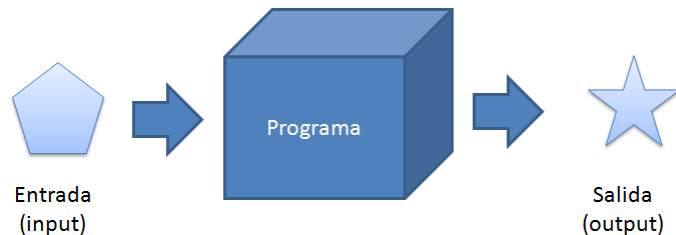
Ambientes – *Determinístico* vs. *no-determinístico*

- Un ambiente determinístico es aquel en el cual una acción tiene un solo efecto garantizado — no hay incertidumbre respecto al estado resultante de efectuar una acción
- El mundo físico puede ser clasificado para todos los propósitos como no-determinístico
- Ambientes no-determinísticos presentan mayores problemas para los diseñadores de agentes



Ambientes - *Episódicos* vs. *no-episódicos*

- En un ambiente episódico, el rendimiento de un agente es dependiente de un número discreto de episodios, sin ninguna dependencia entre la secuencia de estados o episodios
- Ambientes episódicos son más fáciles de manejar, ya que el agente puede decidir qué acción ejecutar basado solo en el episodio actual – no debe razonar acerca de las interacciones entre el episodio actual y los futuros



Ambientes - *Estáticos* vs. *dinámicos*

- Un ambiente estático puede asumirse que no variará, excepto por las acciones del agente
- Un ambiente dinámico tiene varios procesos en operación y que varía en formas que están más allá del control del agente
- Otros procesos pueden interferir con las acciones del agente (como en sistemas concurrentes)
- El mundo físico es un ambiente altamente dinámico



Ambientes – *Discretos vs. continuos*

- Un ambiente es discreto si existe un número fijo y finito de acciones y percepciones de este
- Russell y Norvig citan un juego de ajedrez como ejemplo de un ambiente discreto y manejar un taxi como ejemplo de uno continuo
- Ambientes continuos tienen cierto grado de incompatibilidad con sistemas computacionales
- Ambiente discretos pueden manejarse, en principio, usando una tabla de valores (*lookup table*)

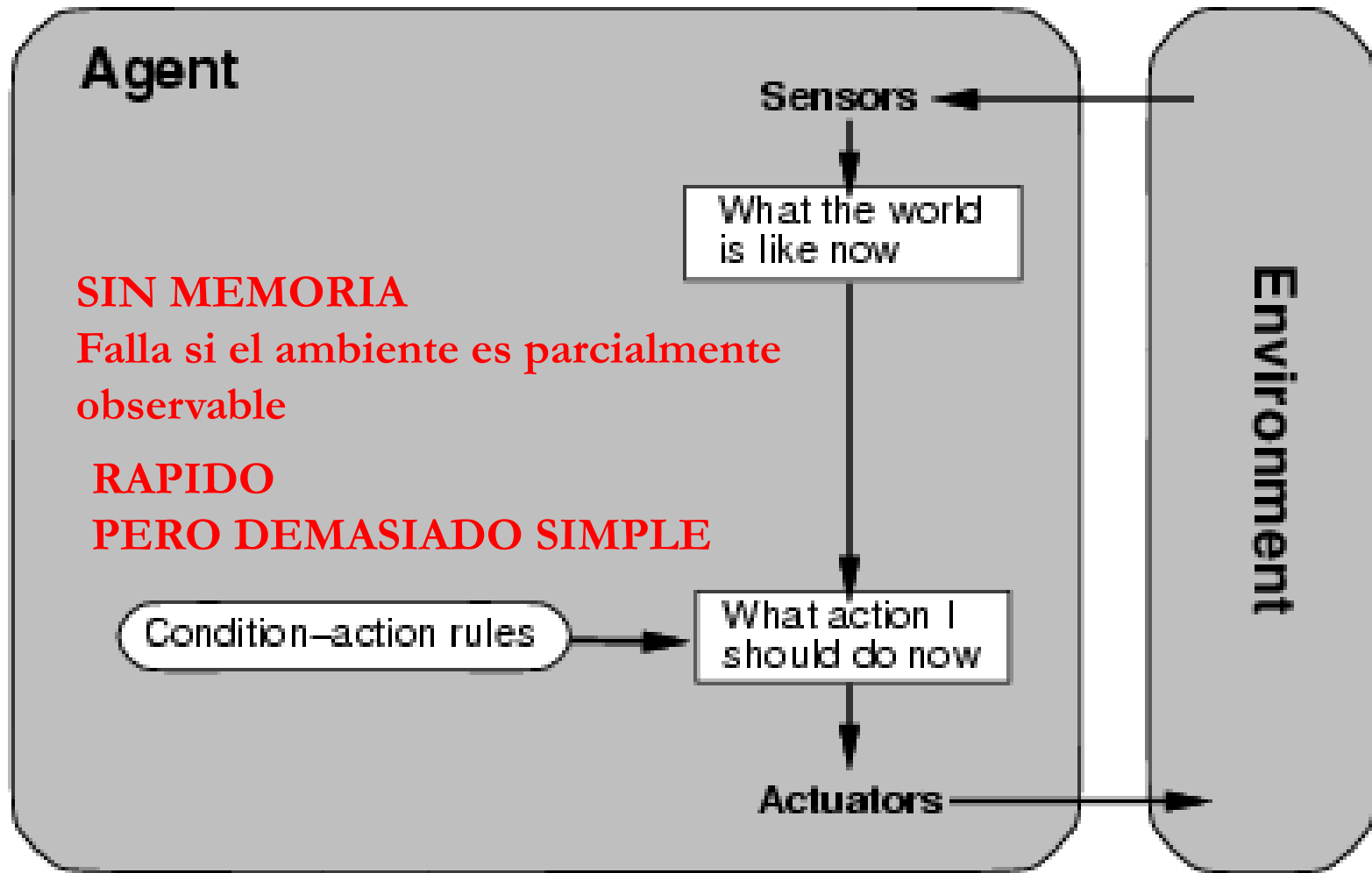


Ejercicio práctico: Ambientes

- Ejemplo: Agente = robot aspiradora (Roomba © y Braava ©)
- Environment (clasificación):
 - *Accesible vs. inaccesible*
 - *Determinístico vs. no-determinístico*
 - *Episódico vs. no-episódico*
 - *Estático vs. dinámico*
 - *Discreto vs. continuo*



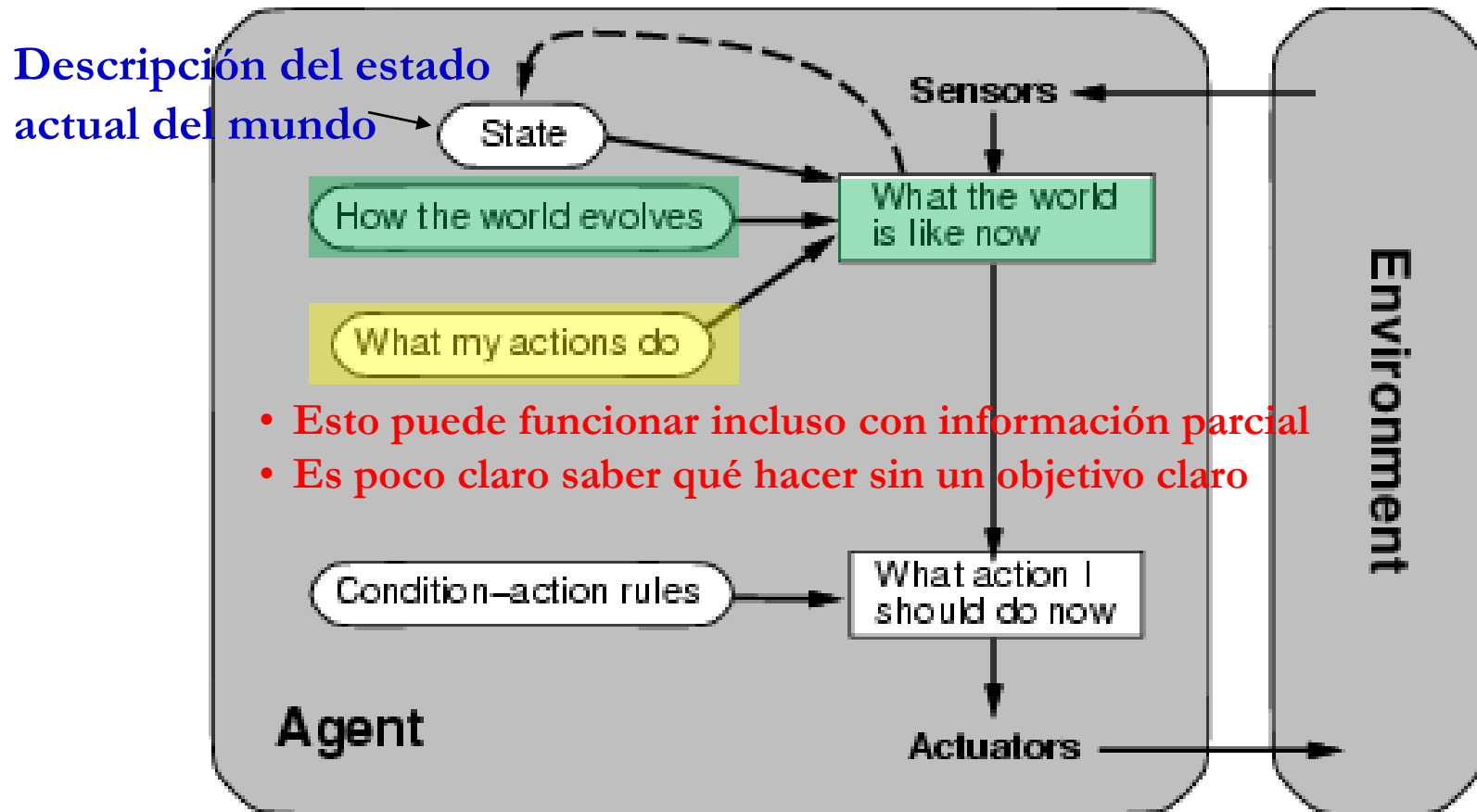
Agentes Reflexivos Simples (Simple reflex)



Agentes Reflexivos Basados en Modelo

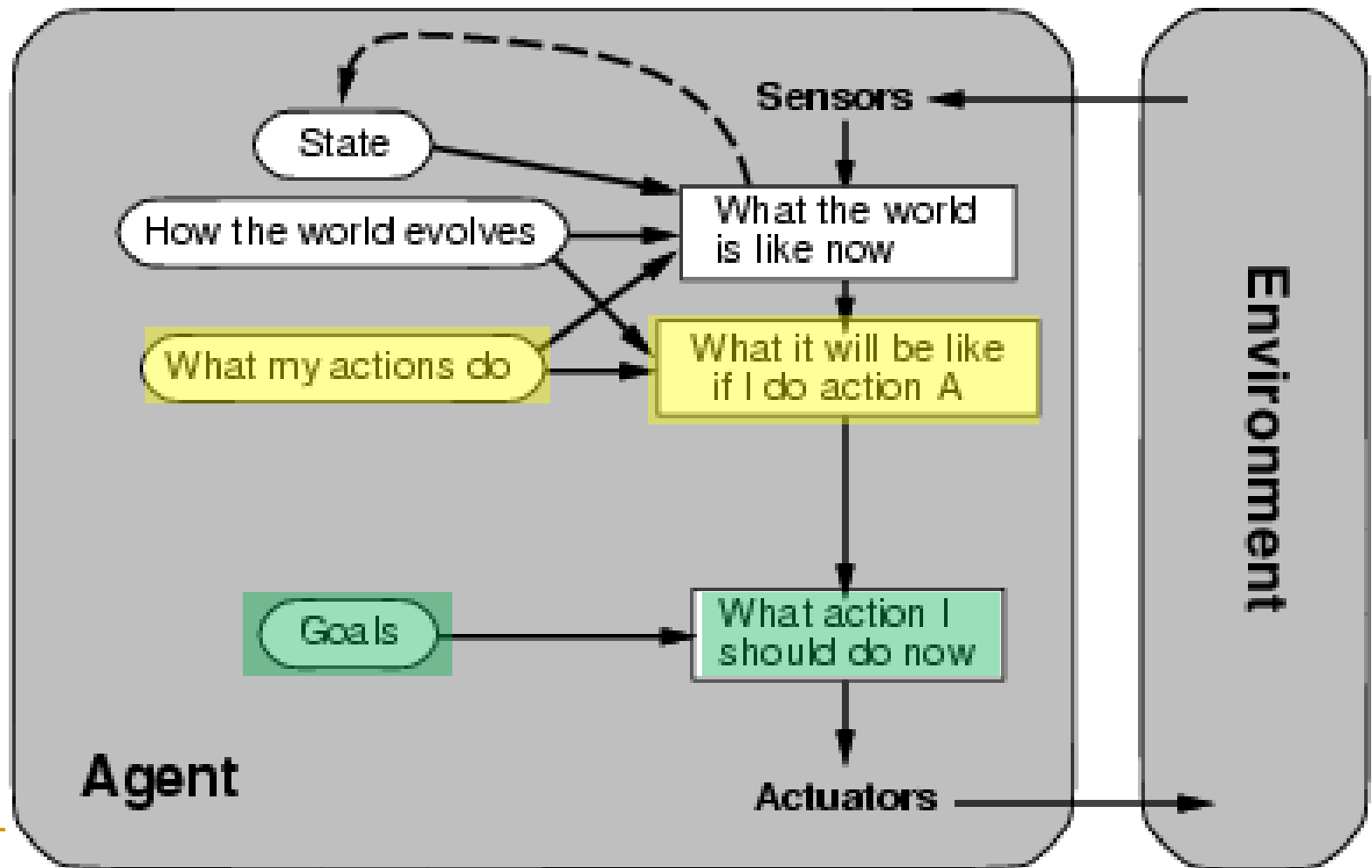
Modela el estado del mundo al:

- Modelar cómo cambia el mundo
- Cómo sus acciones cambian al mundo



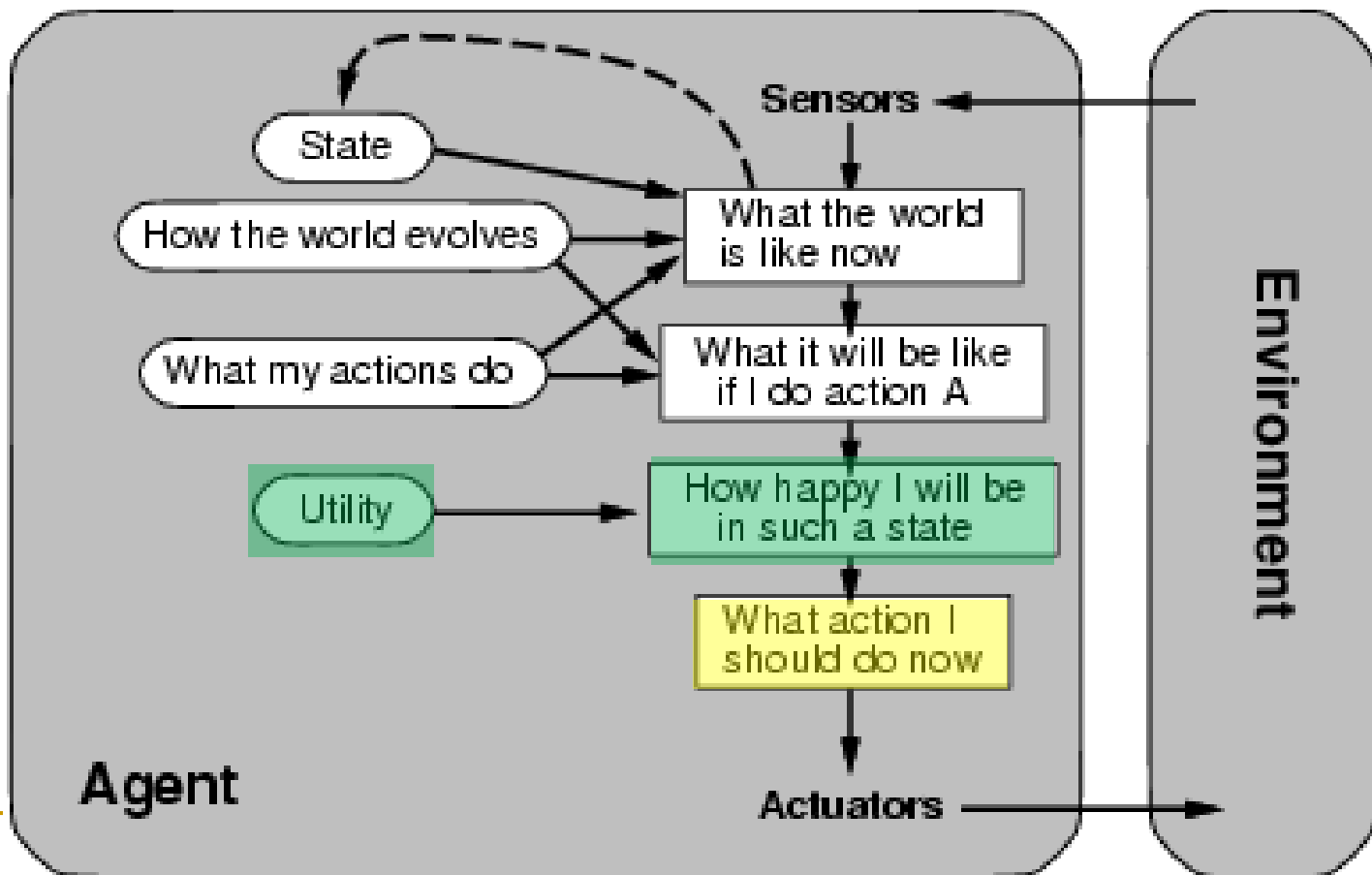
Agentes Basados en Metas

- Objetivos proveen razón para preferir una acción sobre otra.
- Necesitamos predecir el futuro: planificar y buscar.



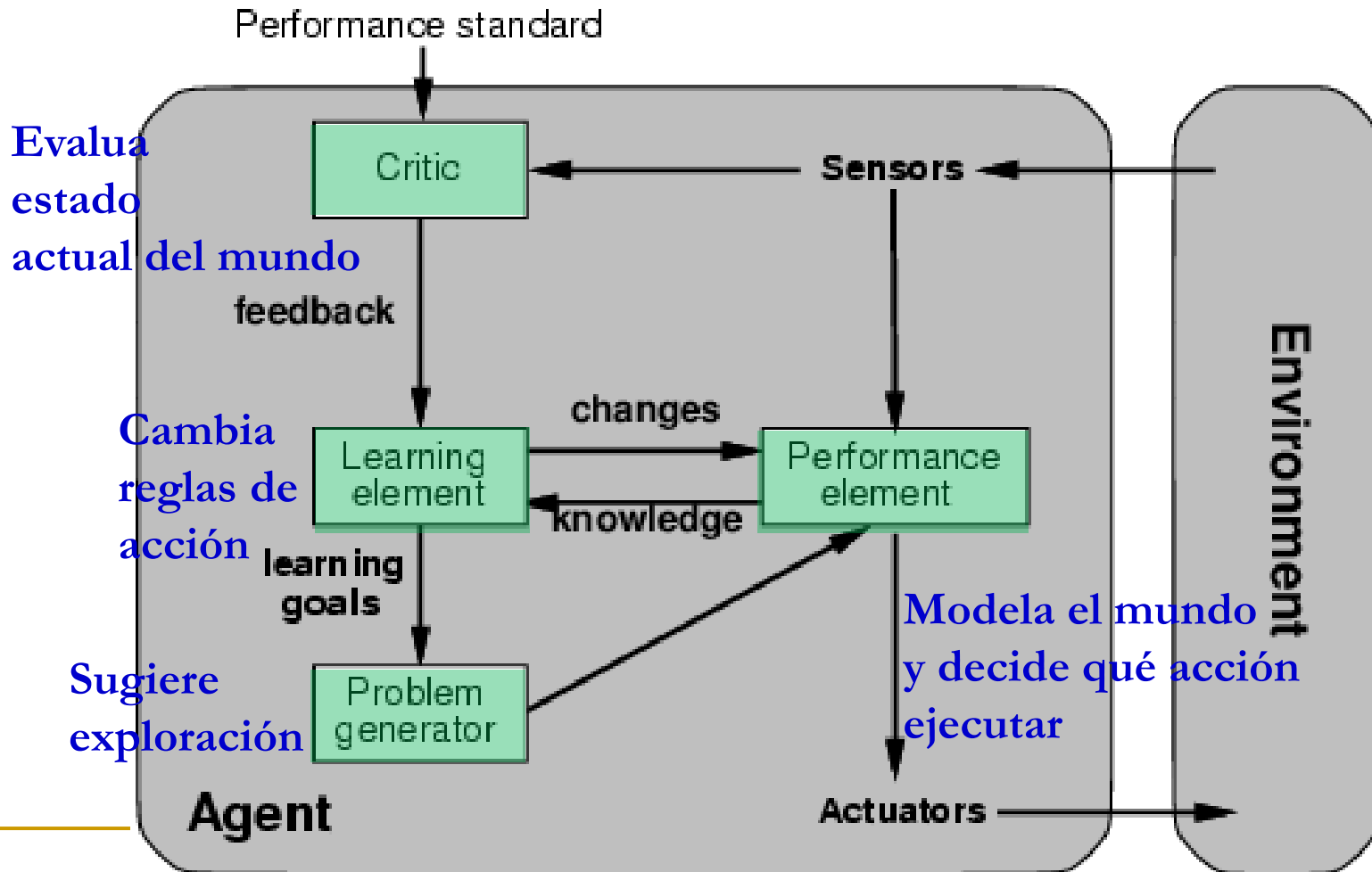
Agentes Basados en Utilidad

- Algunas soluciones para llegar a estados objetivo son mejores que otras.
- Cuál es mejor es determinado por una función de utilidad.
- ¿Qué combinación de objetivos se prefiere?



Agentes que aprenden

- ¿Cómo puede un agente mejorar a través del tiempo?
- Al monitorear su rendimiento y sugerir mejores modelos, nuevas reglas de acción, etc.



Ejercicio práctico: Tipo de agente

- Ejemplo: Agente = robot aspiradora (Roomba y Braava © iRobot)

Performance measure:

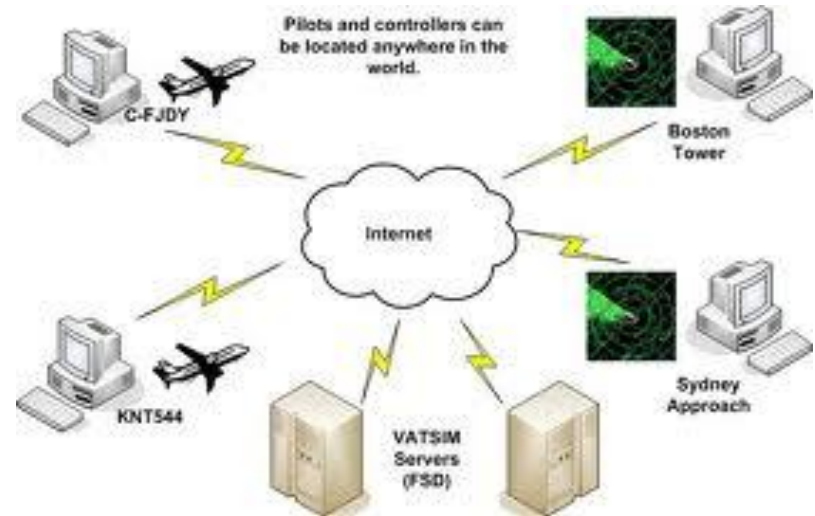
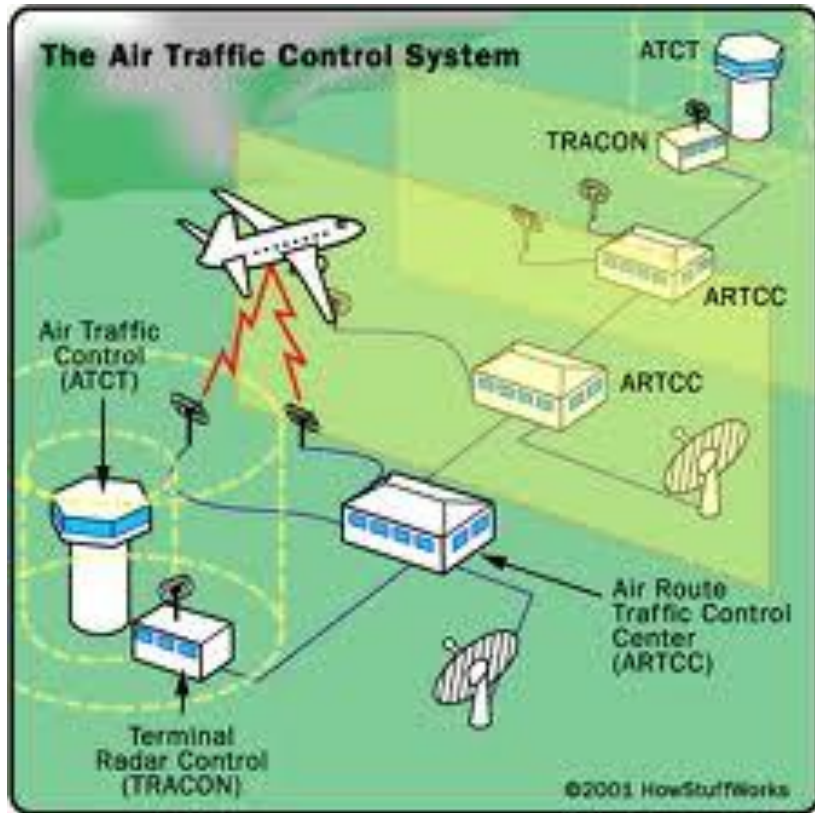
Tipo agente	Tipo de ambiente				
	Accesible Inaccesible	Determinístico No determin.	Episódico No Episódico	Estático Dinámico	Discreto Continuo
Reflexivo simple					
Reflexivo Basado en Modelo					
Basado en Metas					
Basado en Utilidad					
Agente que aprende					

Sistema Multiagentes, una definición

- Un sistema multiagentes consiste en un número de agentes que *interactúan* entre ellos
- En el caso más general, los agentes actuarán en nombre de usuarios con diferentes objetivos y motivaciones
- Para interactuar exitosamente, ellos requieren la capacidad para *cooperar*, *coordinarse*, y *negociar* entre ellos, tal cual lo hacen las personas

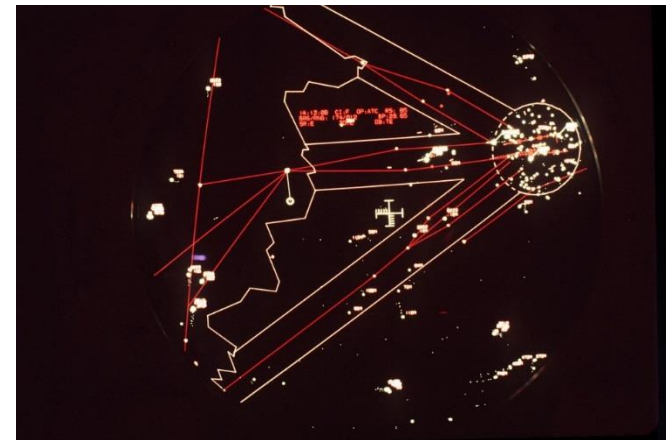
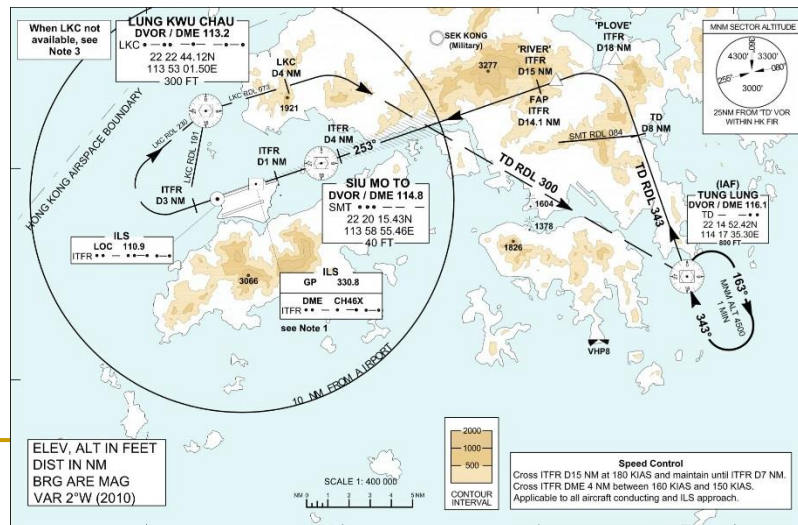


Control de Tráfico Aéreo



Control tráfico aéreo de Sidney

- “Un sistema de control aéreo falla imprevistamente, dejando vuelos sin su soporte. Afortunadamente, sistemas de CA en aeropuertos vecinos reconocen la falla de su par, y cooperan para monitorear y manejar los vuelos afectados”
- Sistemas toman la iniciativa cuando sea necesario
- Agentes cooperan para resolver un problema que va más allá de la capacidad individual de cada agente



Problemas

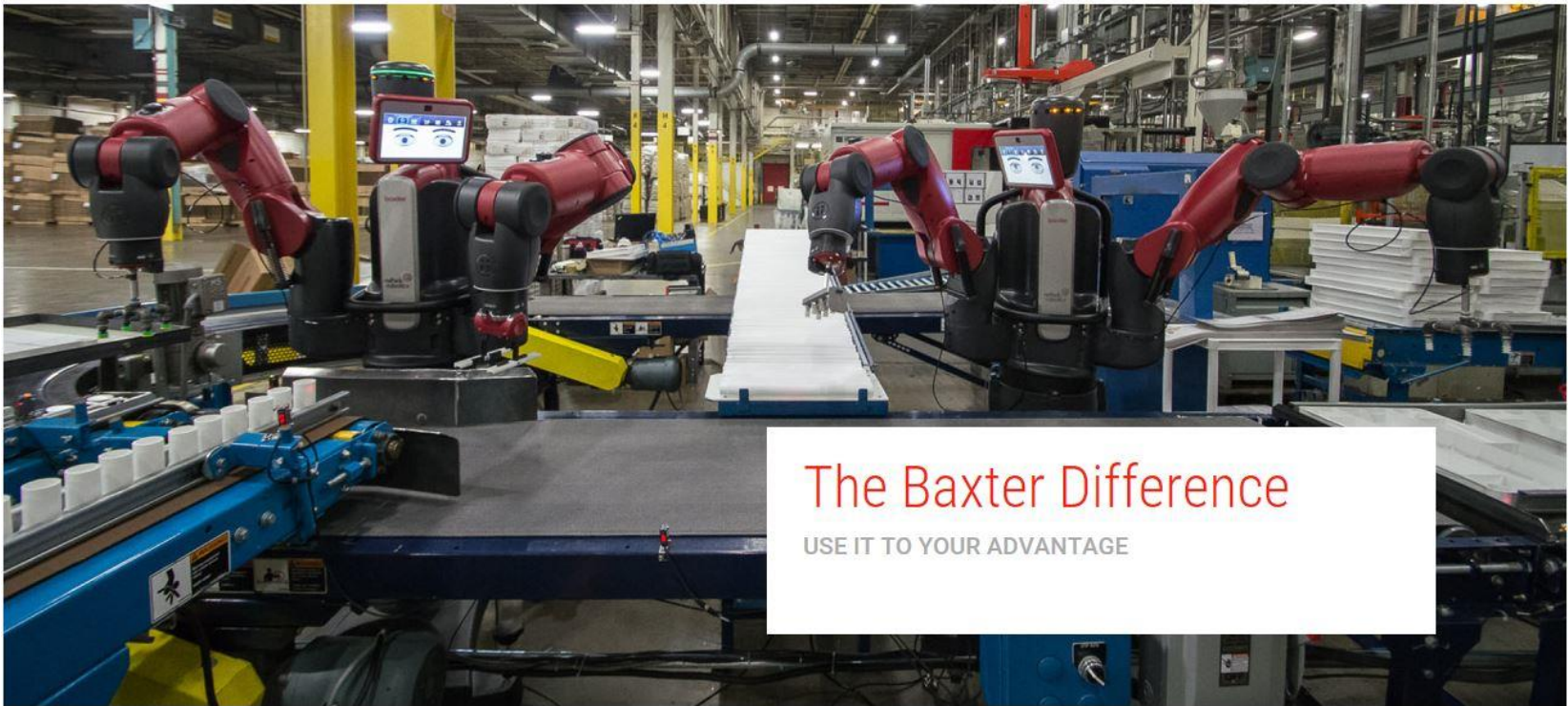
- ¿Cómo manifestar las preferencias a un agente?
- ¿Cómo puede un agente comparar diferentes tratos con diferentes partes?
¿Y si hay muchos parámetros diferentes?
- ¿Qué mecanismos de negociación puede usar un agente para lograr un buen trato con otros agentes?
- Estos problemas son difíciles e importantes – p. ej. abastecimiento automático podría ser usado masivamente por organismos gubernamentales

Sistemas Multiagentes son Interdisciplinarios

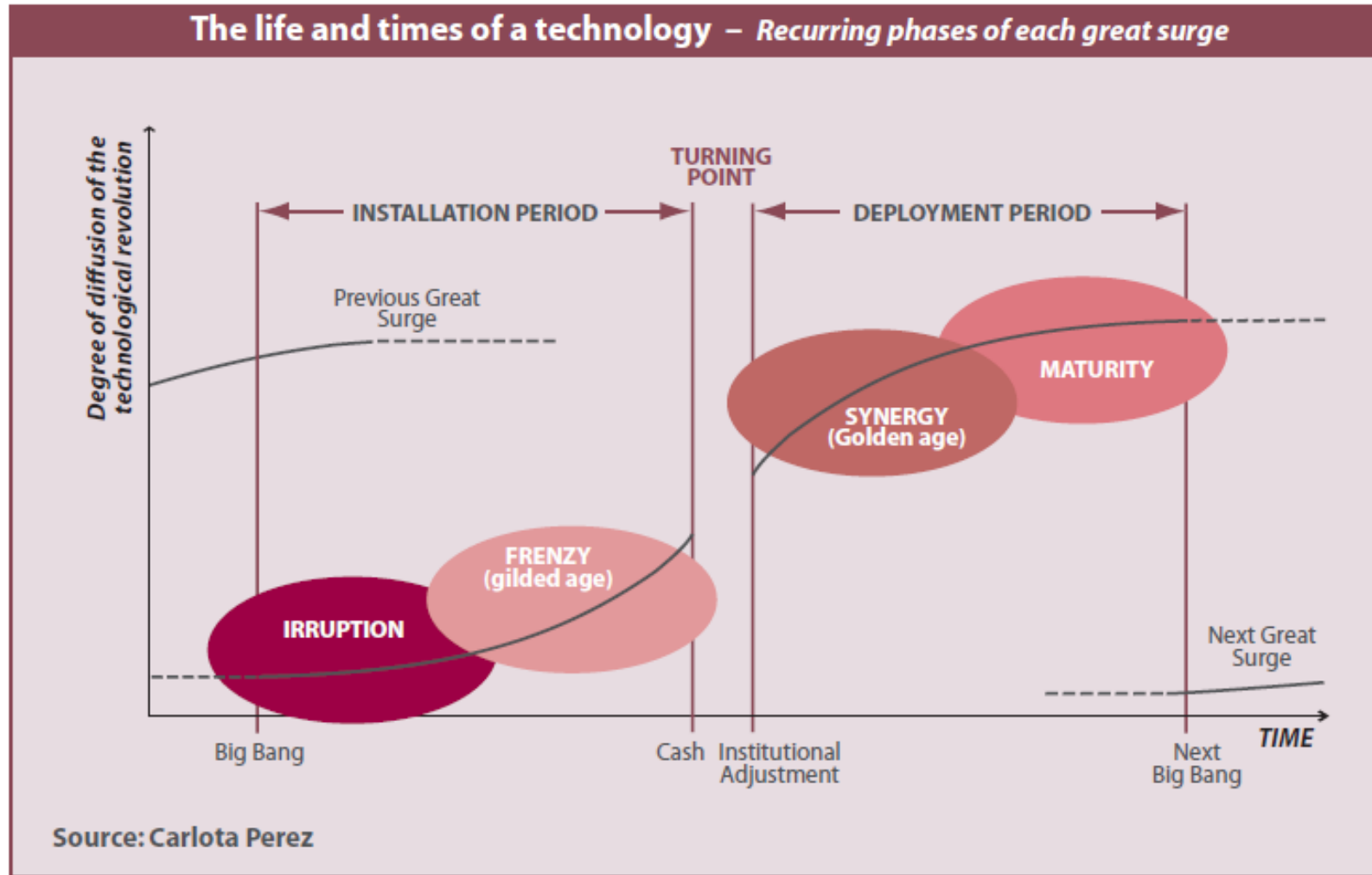
- Inspirados e influenciados por diferentes campos:
 - Economía
 - Filosofía
 - Teoría de juegos
 - Lógica
 - Ecología
 - Ciencias Sociales
- Esto puede ser una fortaleza (aporta metodologías probadas a SMA) y una debilidad (existen muchas visiones de lo que son los SMA)
- Esto es análogo a lo que sucede con la Inteligencia Artificial (IA)

Sistemas de agentes con usuarios

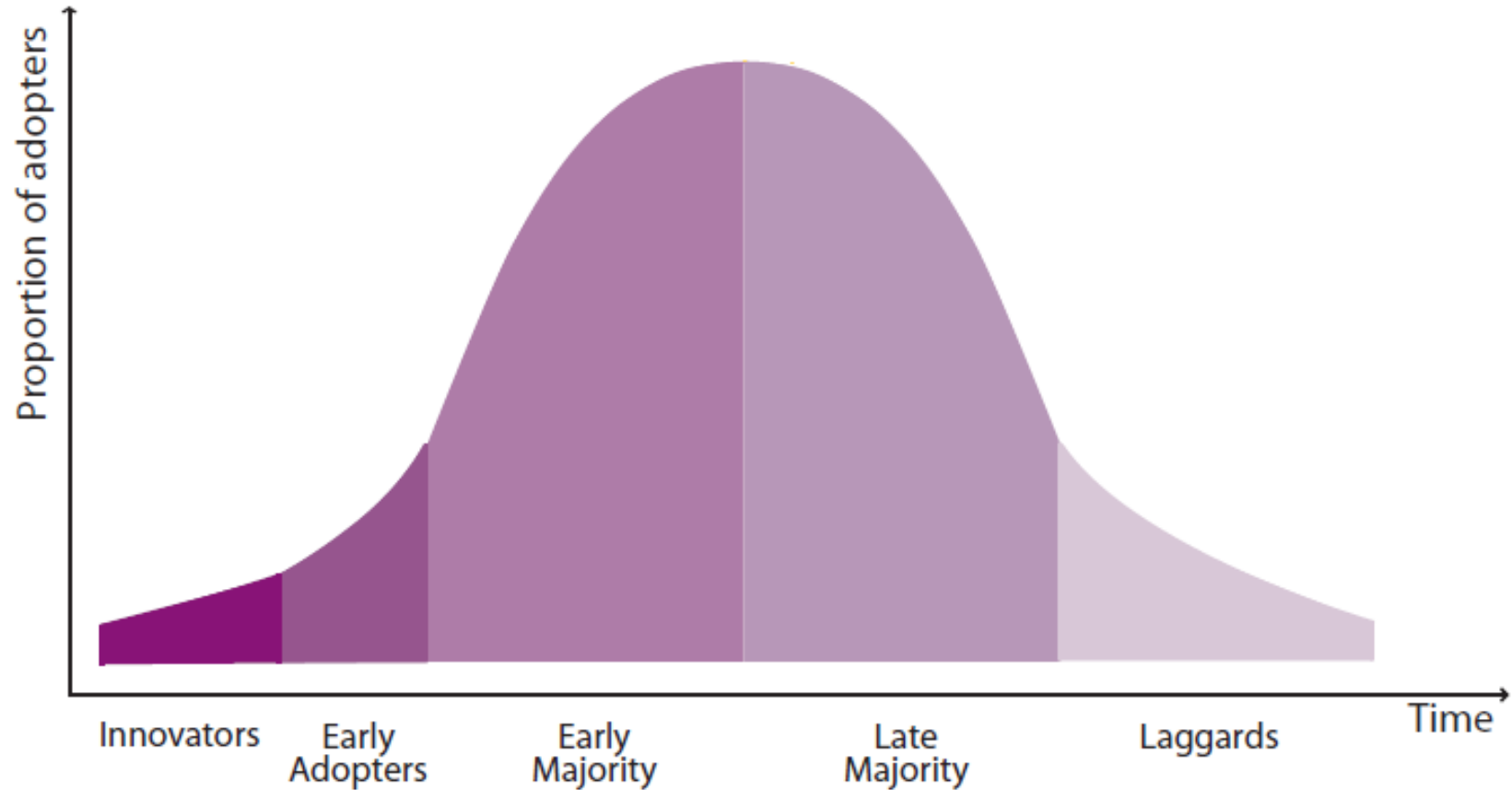
■ Baxter y Sawyer © Rethink Robotics



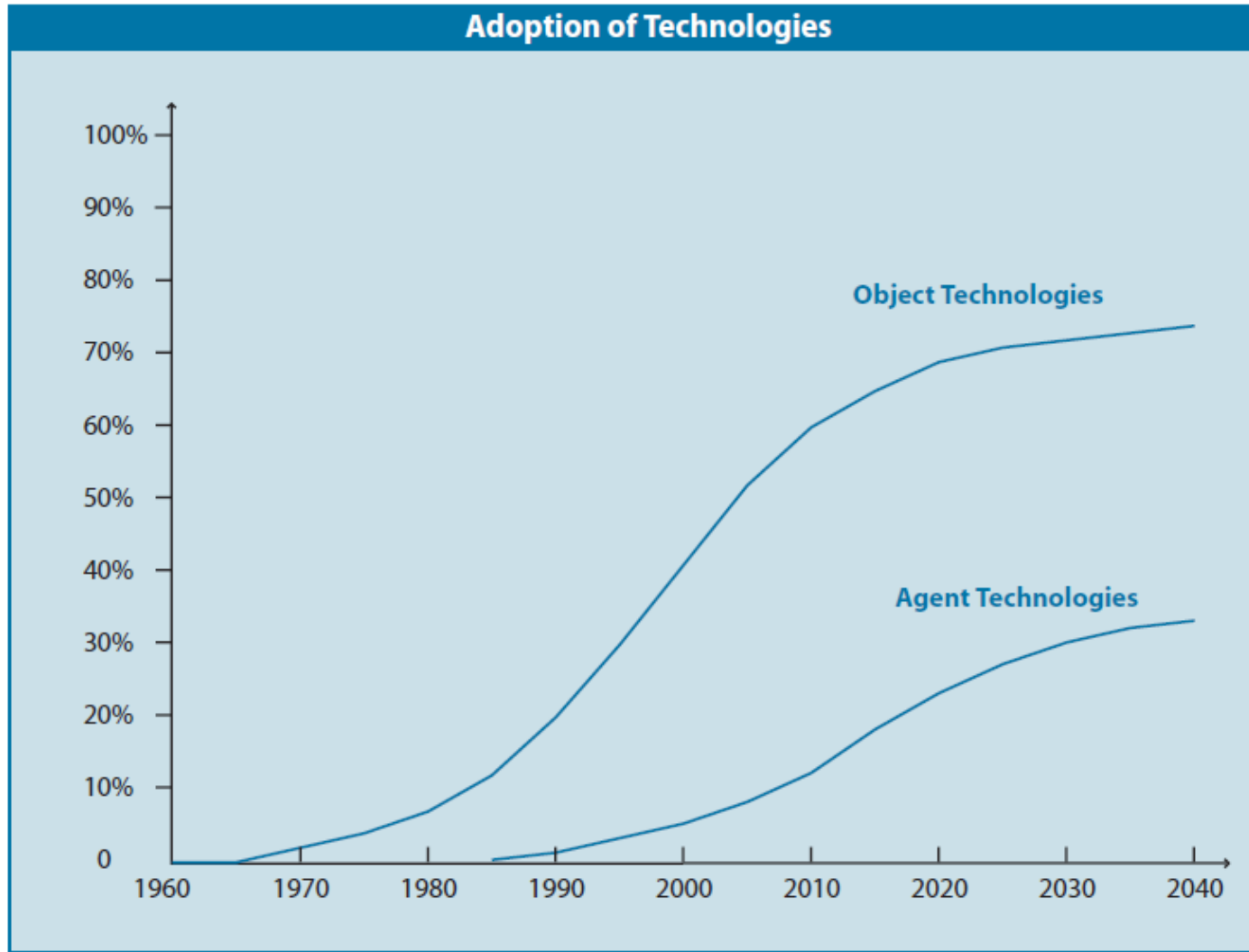
Evolución de sistemas de agentes



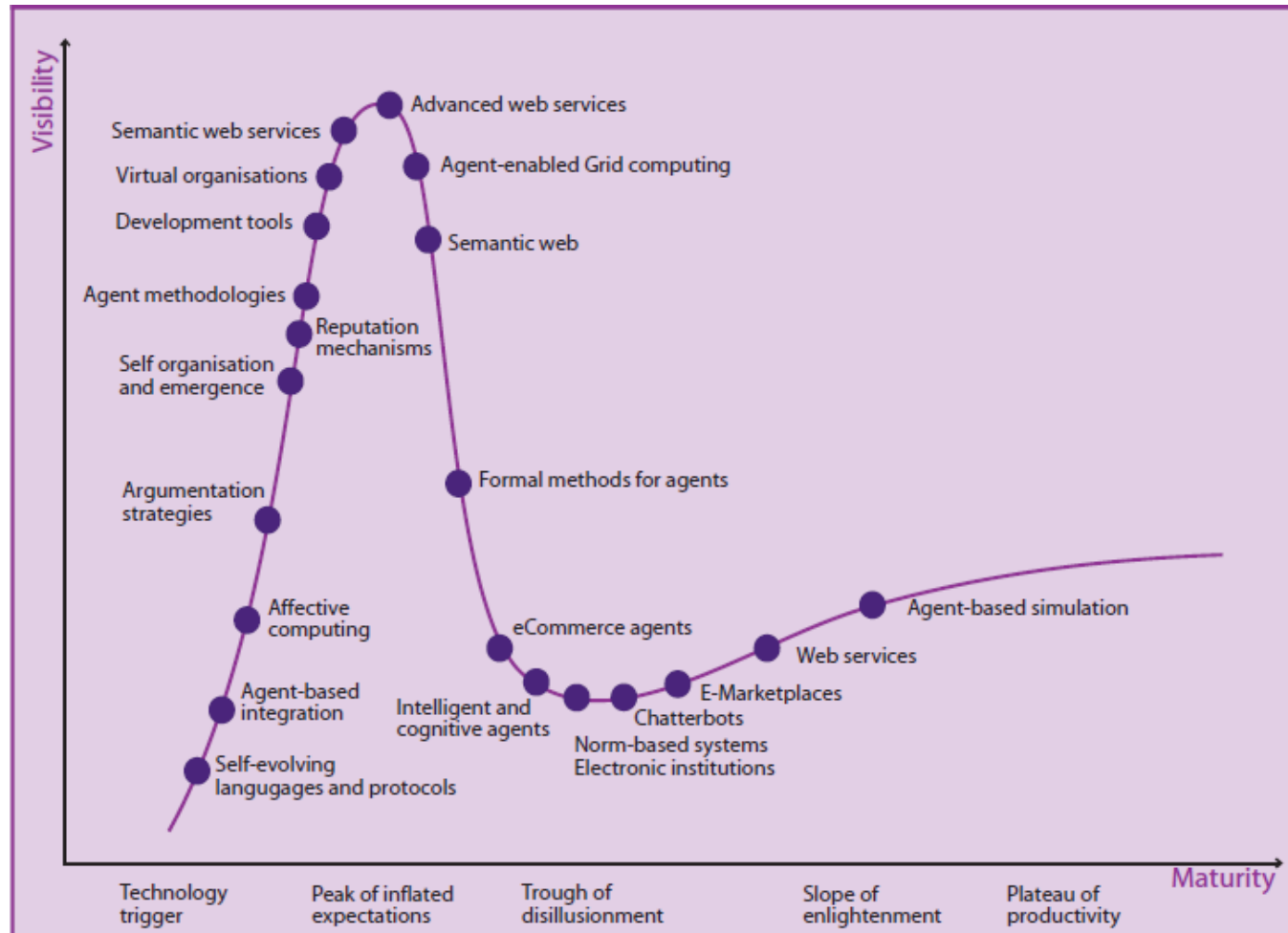
Evolución de sistemas de agentes



Evolución de sistemas de agentes

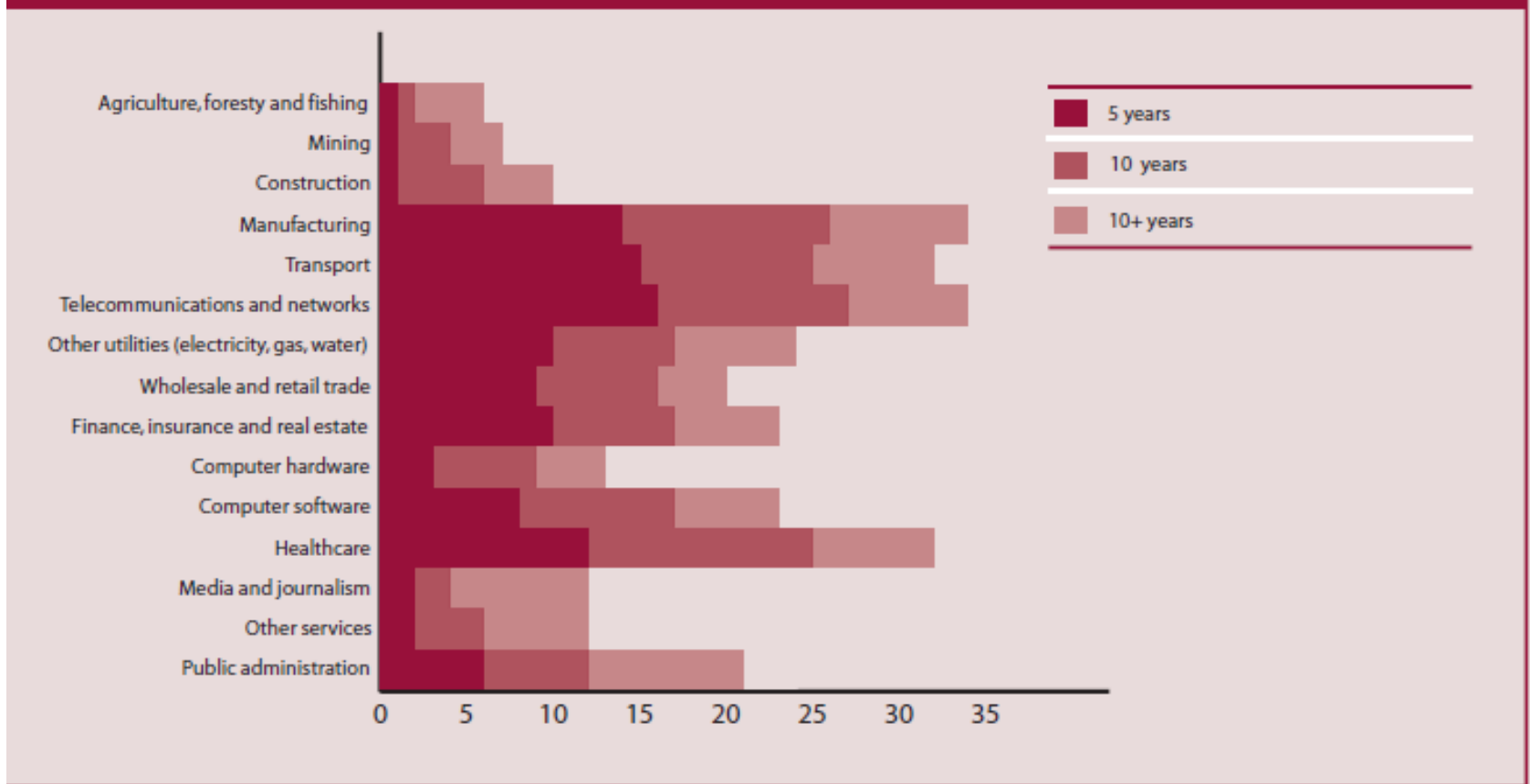


Evolución de sistemas de agentes



Evolución de sistemas de agentes

Figure 6.1: Manufacturing, transport, telecoms and healthcare will encourage agent deployment



Evolución de sistemas de agentes

Figure 6.2: Agents will make the greatest impact in telecoms, transport and manufacturing

