

Fundamentos en Robótica

Unidad 1.2

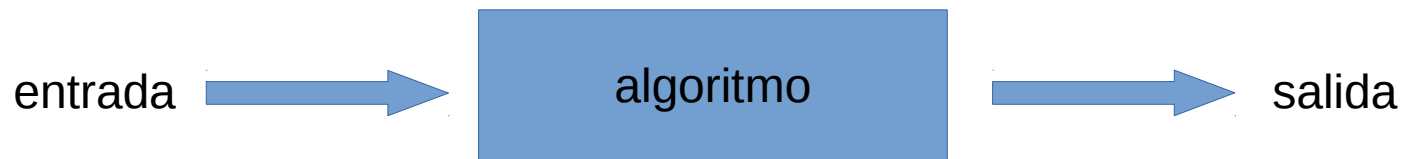
Introducción a la Robótica

Temario

- Historia
- Definiciones
- Componentes
- Categorías
- Aplicaciones
- Inteligencia Artificial
- Paradigmas en robótica

¿Qué es un programa?

Para ciertos datos de entrada (*input*) el programa aplica un algoritmo y genera una salida (*output*).



Los algoritmos son el objeto de estudio de la programación.

¿Qué es un programa?

(Real Academia Española. www.rae.es)

algoritmo.

1. m. Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

Características:

- Preciso: orden en que se realizan los pasos.
- Definido: siempre se obtiene el mismo resultado sin importar el número de veces que se aplique.
- Finito: tiene fin.

¿Qué es un programa?

Ejemplo: encontrar el mayor número de una lista de números positivos, no vacía.

Leo lista

mayor=0

Para cada valor **en** lista

si valor>mayor **entonces**

mayor=valor

Devuelvo mayor

¿Qué es un programa?

Ejemplos:

- Encontrar las raíces de un polinomio de 2º grado.
- Ordenar una lista de números.
- Encontrar la salida de un laberinto.
- Escribir un poema.
- Traducir de un idioma a otro.
- Jugar al ajedrez.
- Ganar al ajedrez.

¿Qué es un programa?

A tener en cuenta:

- Los datos que vamos a tener disponibles a la entrada es parte de la descripción del problema.
- El algoritmo debe garantizar que termina, pero no se dice cuánto tarda.
- Problemas muy complejos llevan a aplicar Inteligencia Artificial.

Historia de la I.A.

- <1950
 - Inteligente = recuerda, hace cuentas
 - (modelos de neuronas, tortuga de Grey)
- 1950-1970
 - Inteligente = aplica lógica, resuelve problemas
 - Optimismo, discusiones teóricas, Test de Turing
 - (redes neuronales, perceptrones)
- 1980s
 - Sistemas expertos, “5ª generación”
- 1990-
 - Lenguaje natural, data-mining, aprendizaje, métodos probabilísticos

¿Qué es IA?

Pensar como un humano

- Ciencias cognitivas
- Procesamiento visual

Pensar racionalmente

- Lógica
- Inferencia, demostración

Actuar como un humano

- Test de Turing
- Procesamiento de lenguaje natural

Actuar racionalmente

- Obtener buenos resultados
- Aprendizaje automático, metaheurísticas

Actuar en forma racional

- Agente inteligente
 - Un agente es algo capaz de percibir y actuar.
- Existen maneras de actuar racionalmente que no requieren de inferencia.
- Hay situaciones en las cuales se debe actuar y no existe una forma de determinar lo correcto.
- La racionalidad de un agente se mide evaluando las consecuencias de su acciones.
 - Se necesita una métrica sobre los estados del entorno

Fundamentos de la IA

Hereda ideas, puntos de vista y técnicas:

- Filosofía
- Matemáticas
- Psicología
- Lingüística
- Computación
- Economía
- Neurociencias
- Cibernética

Las siete áreas de IA

- Representación de conocimiento.
- Entendimiento del lenguaje natural.
- Aprendizaje.
- Planificación y resolución de problemas.
- Inferencia.
- Búsqueda.
- Visión.

Especificando un agente

Describiendo un agente:

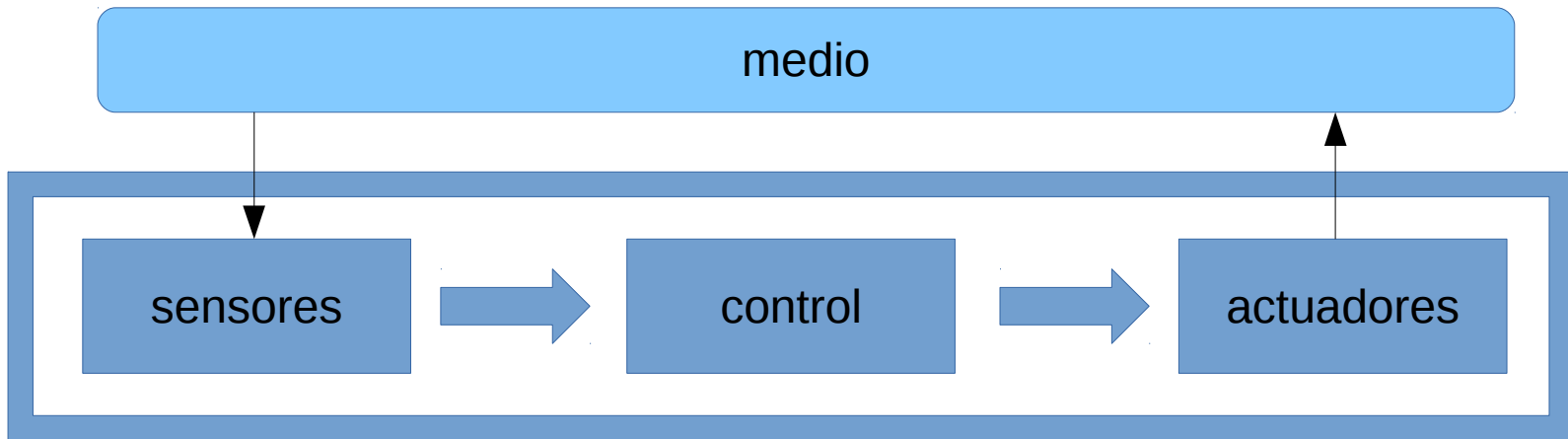
- Métrica de rendimiento (*performance*)
- Entorno
- Actuadores
- Sensores

Especificando un agente

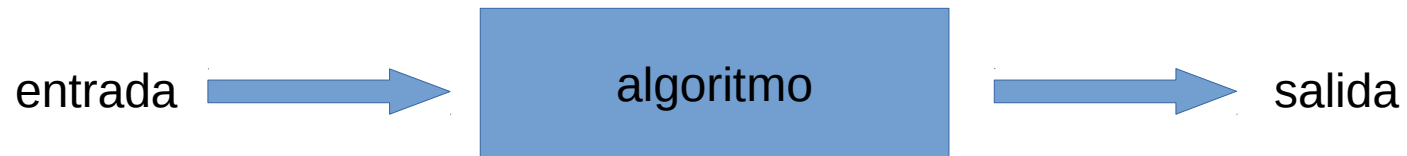
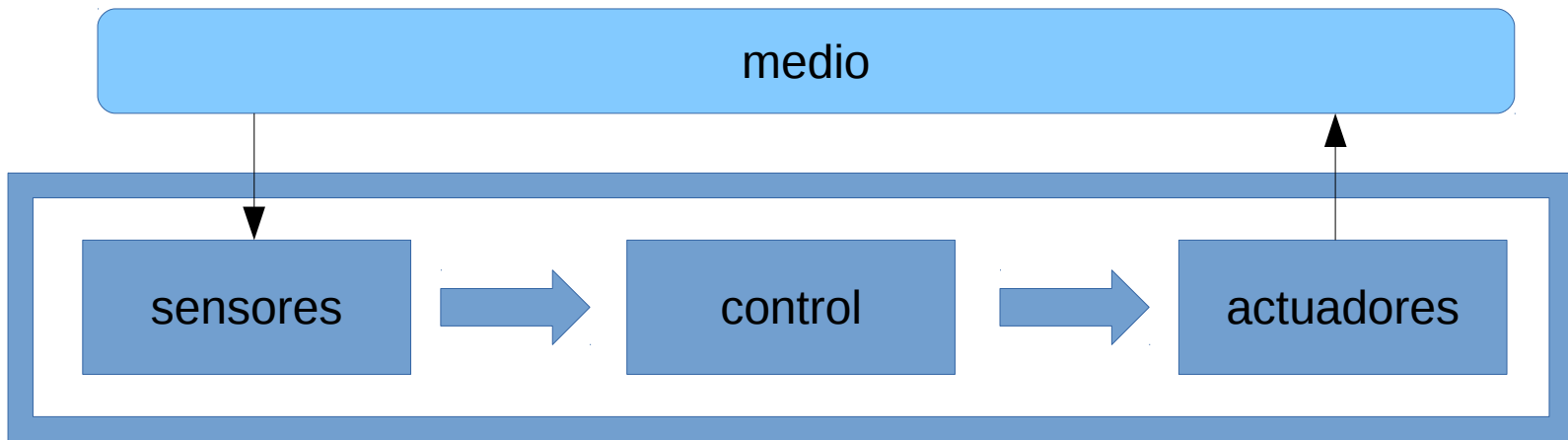
Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry

Figure 2.5 Examples of agent types and their PEAS descriptions.

Entorno y agente



Control



Paradigmas

“Conjunto de asunciones y/o técnicas que caracterizan una aproximación a una clase de problema.”

En robótica, difieren en cómo usan y el rol que le asignan a las funciones primitivas:

- Sensar (SENSE)
- Planificar (PLAN)
- Actuar (ACT)

Paradigmas

Primitiva robótica	Entrada	Salida
Sensar (SENSE)	Datos de los sensores	Información sensada
Planificar (PLAN)	Información (sensorial o cognitiva)	Directivas
Actuar (ACT)	Información sensada o directivas	Comandos a los actuadores

Paradigmas

Existen tres paradigmas para organizar la inteligencia en un robot:

- Jerárquico (Deliberativo)
- Reactivo
- Híbrido Deliberativo/Reactivo

Paradigma jerárquico

Robot + IA clásica = Paradigma jerárquico

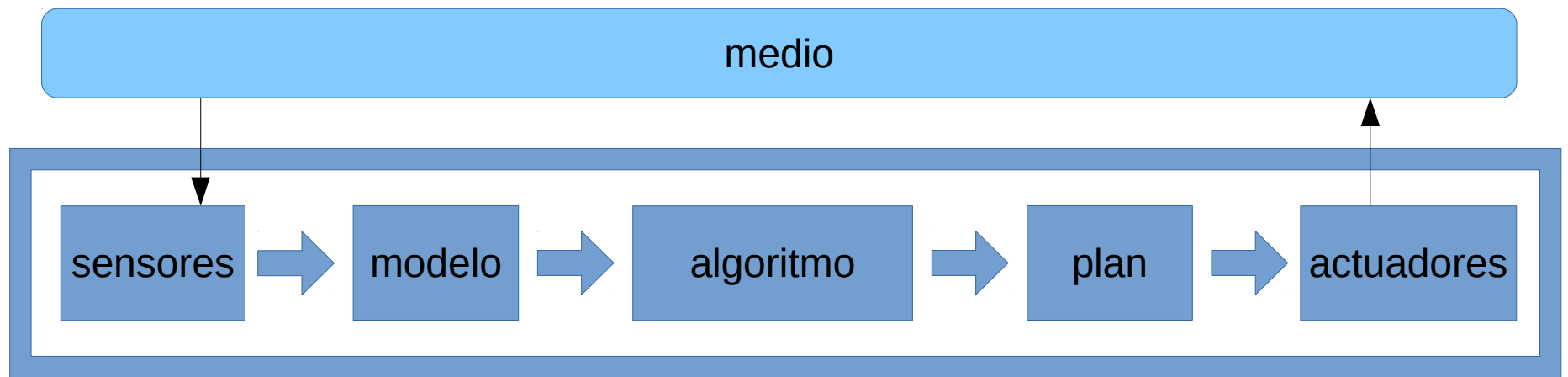
*Sabemos escribir programas que juegan a ajedrez.
Mover un robot no puede ser más complicado.*

(1965 - 1990)

Paradigma jerárquico

Primitiva robótica	Entrada	Salida
Sensar (SENSE)	Datos de los sensores	Información sensada
Planificar (PLAN)	Información (sensorial o cognitiva)	Directivas
Actuar (ACT)	Información sensada o directivas	Comandos a los actuadores

Paradigma jerárquico



Paradigma jerárquico

Aplicación directa de técnicas de Inteligencia Artificial clásica a la robótica:

- Recolecto información, y armo un **modelo** del mundo.
- **Computo** una solución a mi problema sobre el modelo, y determino pasos a seguir para aplicar la solución.
- Ejecuto **plan** de acción.

Miro, cierro los ojos, pienso, hago

Paradigma jerárquico

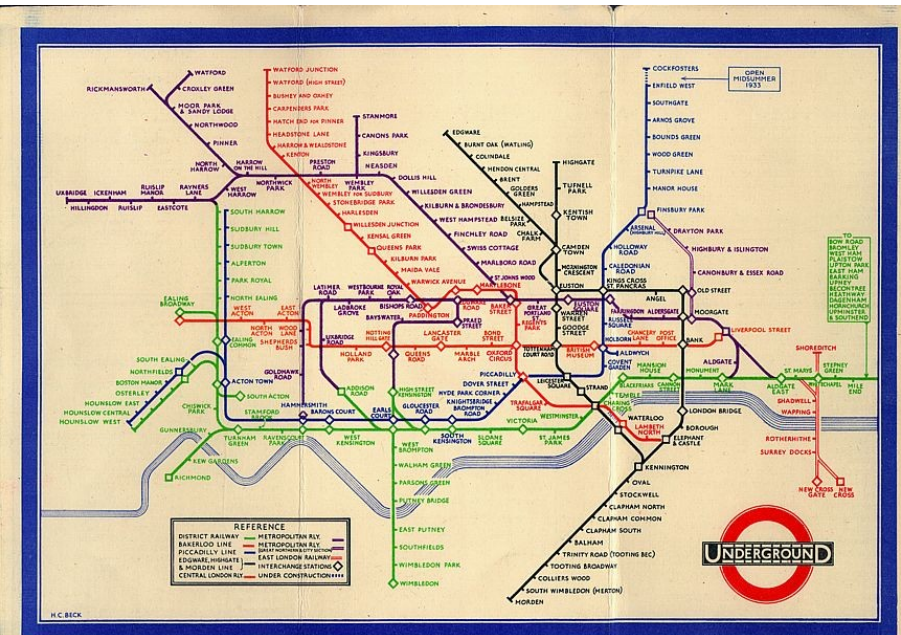
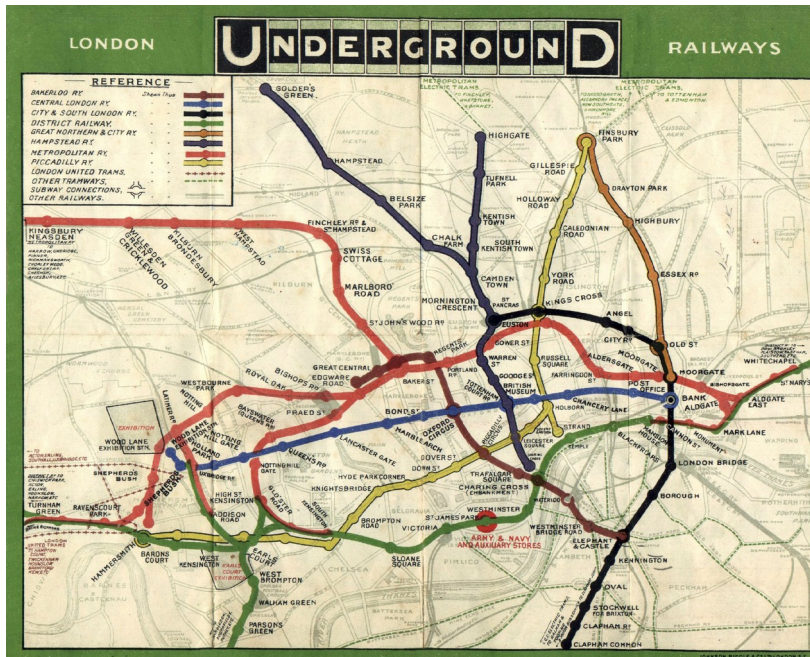
Ejemplo: Shakey (1967-1971)



Paradigma jerárquico

Ejemplo: robot que puede llegar a un destino (habitación, oficina, dirección...)

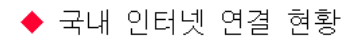
- Mapa métrico (cuantitativo): grilla de ocupación, nube de puntos.
- Mapa topológico (cualitativo): grafo



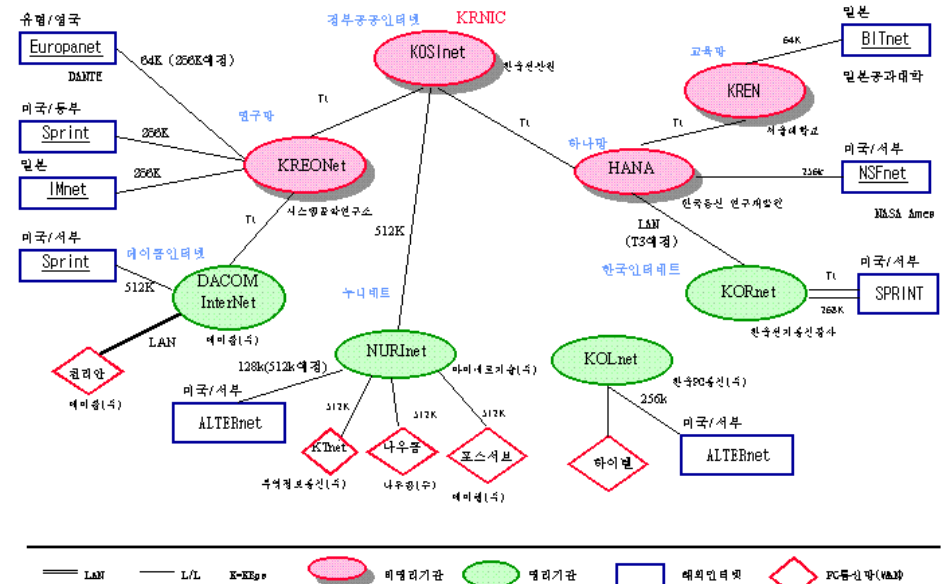
1909

Harry Beck, 1933

Ejemplo: robot que puede llegar a un destino (habitación, oficina, dirección...)

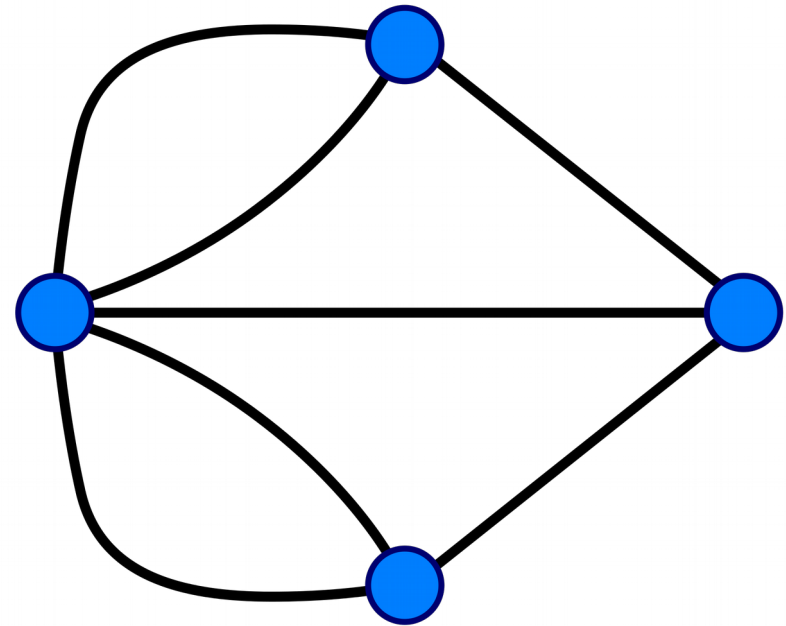


일자 : 1995. 5. 31
작성 : 한국인터넷정보센터



Paradigma jerárquico

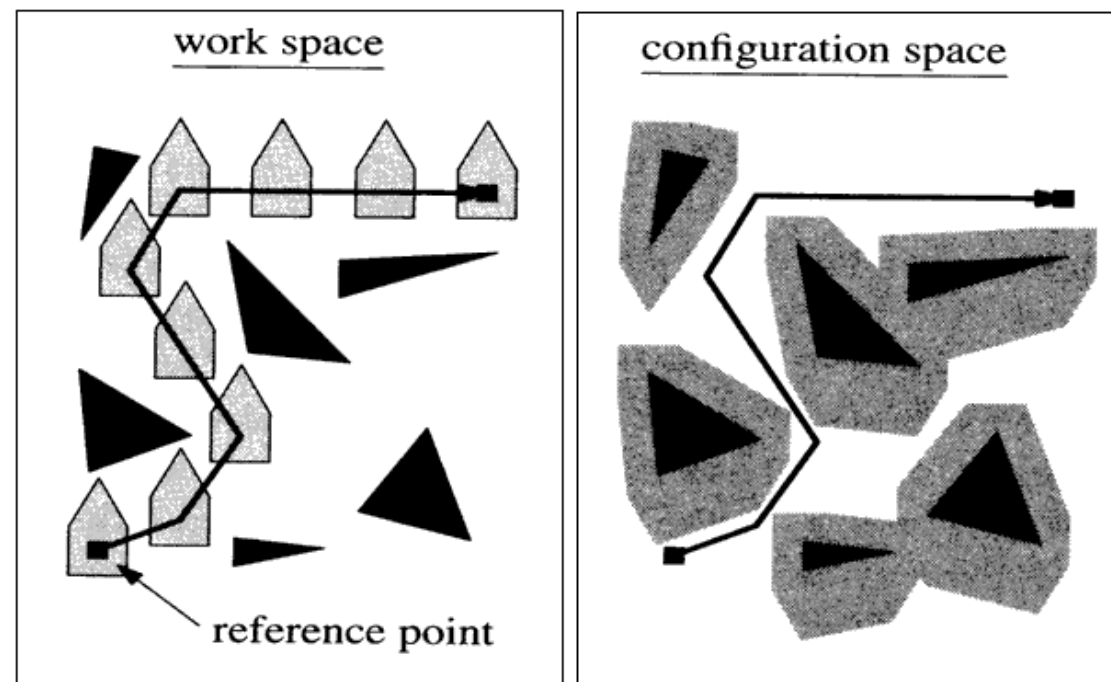
Ejemplo: robot que puede llegar a un destino (habitación, oficina, dirección...)



Paradigma jerárquico

Ejemplo: robot que puede llegar a un destino (habitación, oficina, dirección...)

- Navegación: construcción de camino óptimo.
 - Encontrar camino mas corto en un grafo: Dijkstra, A^* , D^* , Bellman-Ford...
- Espacio de configuración



Paradigma jerárquico

En la práctica, el paradigma jerárquico funciona mucho peor de lo que se podría esperar de sus algoritmos de base: el mundo real no es un tablero de ajedrez.

Jugador de ajedrez - Cadete de oficina

Ajedrez vs. Realidad

¿El entorno es Determinista?

Conociendo el estado actual, es posible predecir exactamente el resultado de aplicar una acción.

Jugador de ajedrez: si (“estratégico”).

Cadete de oficina: no.

Ajedrez vs. Realidad

¿El entorno es Discreto o Continuo?

Se aplica a la representación del estado del entorno, percepción, actuación y tiempo.

Jugador de ajedrez: Discreto.

Cadete de oficina: Continuo

Ajedrez vs. Realidad

¿El entorno es Observable?

El sistema sensorial tiene acceso a toda la información necesaria para representar internamente el estado del entorno.

Jugador de ajedrez: si.

Cadete de oficina: no.

Ajedrez vs. Realidad

¿El entorno es en Tiempo Real?

En un sistema de Tiempo Real, los procesos ocurren en el tiempo físico, externo e independiente. El tiempo no se detiene ni frena, nunca.

Jugador de ajedrez: No*.

Cadete de oficina: Si.

Ajedrez vs. Realidad

¿El entorno es en Episódico o Secuencial?

En un sistema de Episódico la decisión en cada momento es independiente de las decisiones tomadas en otro momentos.

Jugador de ajedrez: Secuencial.

Cadete de oficina: Secuencial.

Ajedrez vs. Realidad

¿El entorno es Estático o Dinámico?

Un entorno dinámico puede cambiar mientras el robot está deliberando.

Jugador de ajedrez: estático*.

Cadete de oficina: dinámico.

Ajedrez vs. Realidad

¿El entorno es Multiagente?

Un entorno multiagente tengo en cuenta a otros agentes con sus propios PEAS

Jugador de ajedrez: multiagente competitivo.

Cadete de oficina: ?

Entornos

Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete
Figure 2.6 Examples of task environments and their characteristics.						

Reglas vs. Leyes

Reglas, sé que las cosas suceden de cierta manera.

- Fútbol: número de arcos.
- 2ª Ley de la Termodinámica.

Leyes, pueden violarse, con consecuencias específicas.

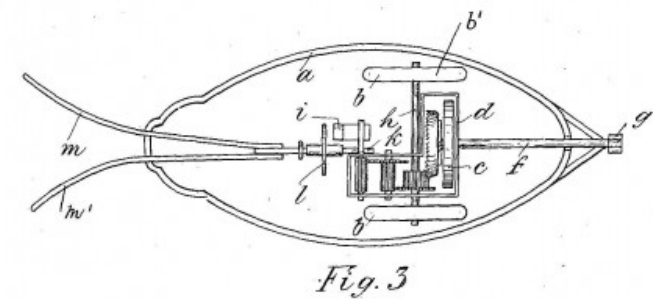
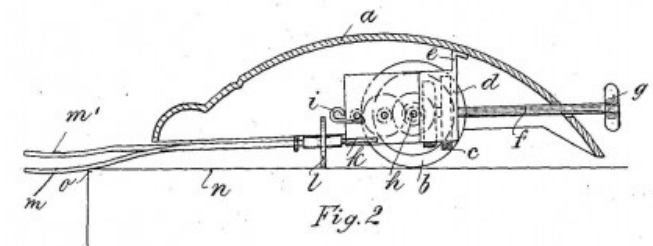
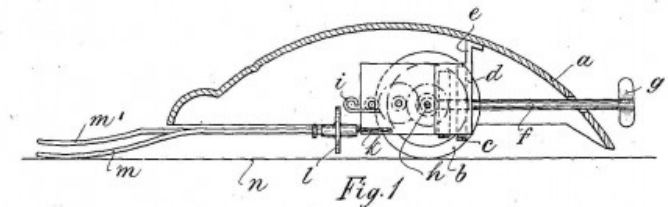
- Fútbol: no tocar la pelota con la mano.
- No matarás.
- “*Violación de la Ley $\Rightarrow$ Consecuencia*” es una regla.

Robots que no piensan

Es posible hacer robots

- Sin computadoras.
- Que no sean inteligentes.
- Que resuelvan problemas.

A. WEIGEL.
MOVING TOY.
APPLICATION FILED MAR. 20, 1911.
1,017,066. Patented Feb. 13, 1912.



Witnesses:
Max Lemmer
Paul Gschlopp

Inventor:

Adolf Weigel

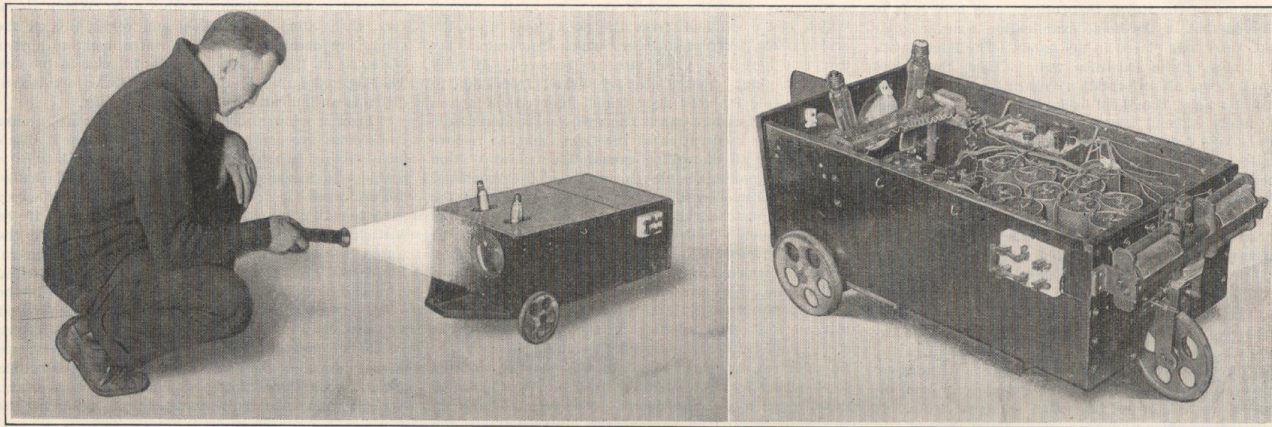
Robots que no piensan

...otro ejemplo.

376

SCIENTIFIC AMERICAN SUPPLEMENT No. 2267

June 14, 1919



The Electric Dog following its luminous master, and a general view of its internal arrangement

The Electric Dog

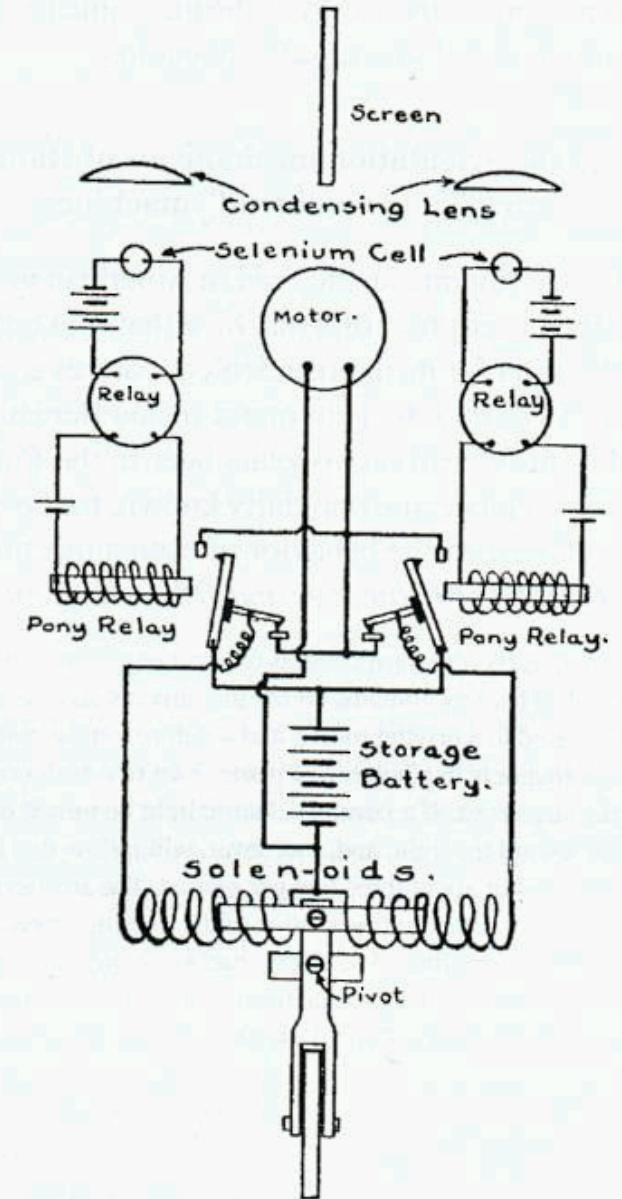
Use of the Selenium Cell to Make an Orientation Mechanism

By Benjamin Franklin Miessner

The principle of orientation is well exemplified in nature. A large variety of plants, flowers, and other living examples of the realm of botany, are endowed with the power of turning themselves for the purpose of receiving the beneficial effect of the sun's rays. A

possessed of its own mechanical brain, my work on the electric dog represents an attempt to evolve an automaton which would conduct itself in much the same manner as do some of the lower forms of animal

fully in rendering an automaton susceptible only to the call of its master is to prevent the calls of others from reaching it. This the orientation mechanism has done for the searchlight-selenium or infra-red ray systems of



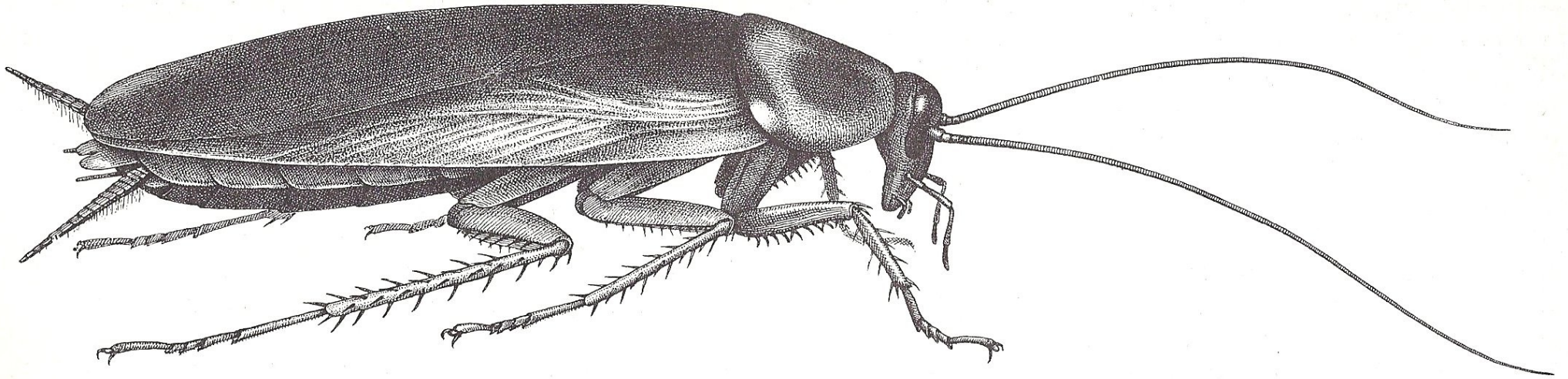
Observando la naturaleza

Inspiración: polilla



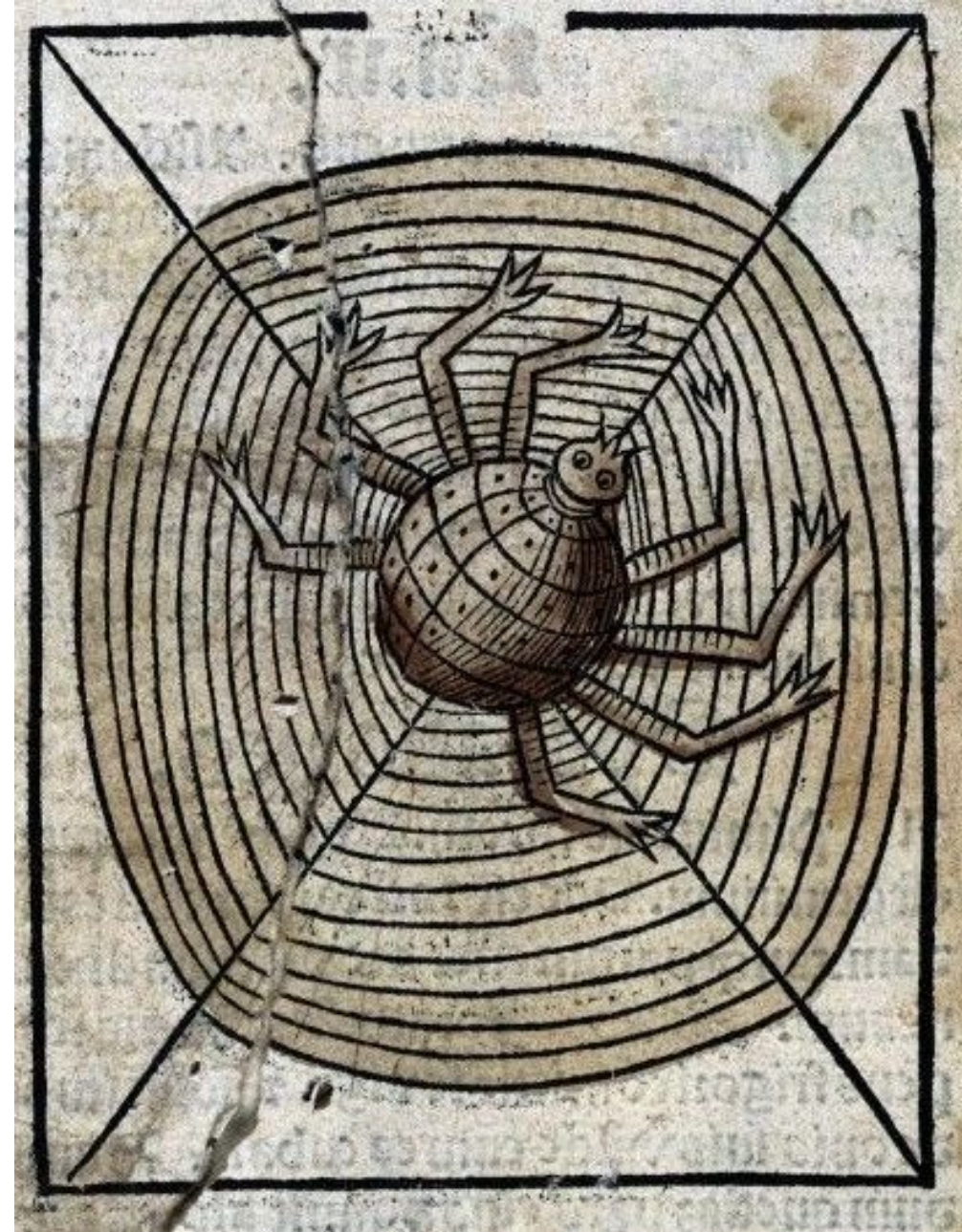
Observando la naturaleza

Inspiración: cucaracha



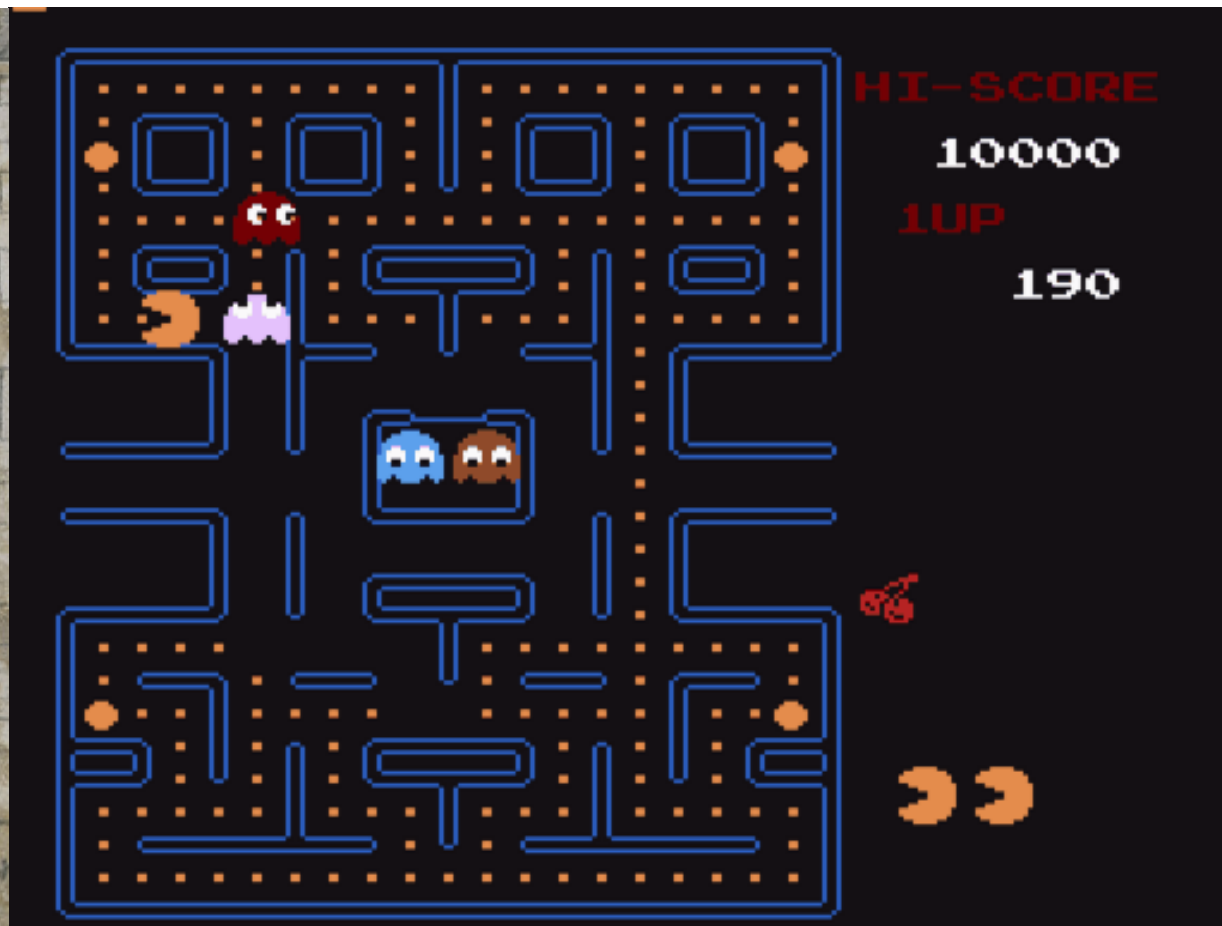
Observando la naturaleza

Inspiración: arañas



Observando la naturaleza

Inspiración: laberintos



Paradigma reactivo

Un comportamiento es una asignación de estímulos sensoriales a un patrón de acción motora que se utiliza para realizar una tarea.



Reflejos - Memoria muscular

Paradigma reactivo

Primitiva robótica	Entrada	Salida
Sensar (SENSE)	Datos de los sensores	Información sensada
Planificar (PLAN)	Información (sensorial o cognitiva)	Directivas
Actuar (ACT)	Información sensada o directivas	Comandos a los actuadores



Paradigma reactivo

- Describir la tarea
- Describir el robot
- Describir el entorno

Especificación y Análisis

- Describir como el robot debería reaccionar al ambiente

Diseño

- Implementar cada comportamiento
- Probar cada comportamiento

Implementación

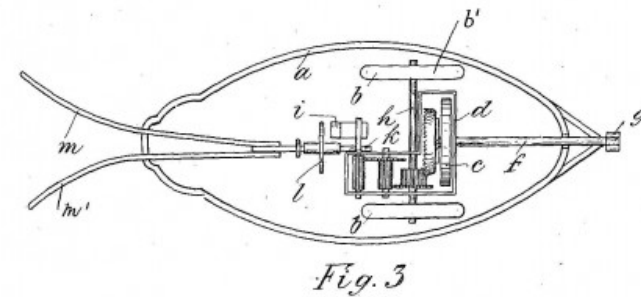
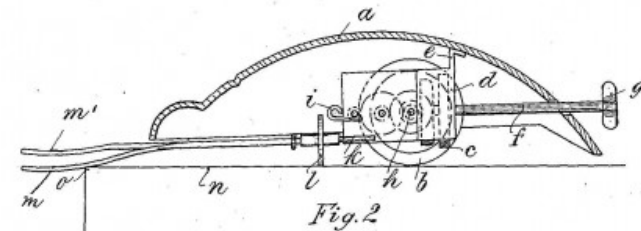
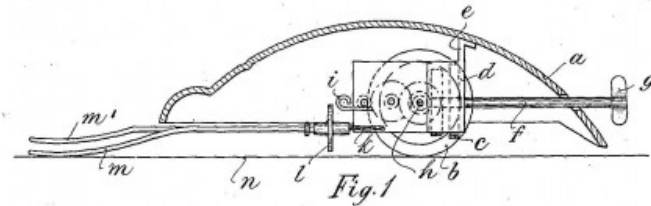
- Juntar todos los comportamientos
- Probar todos lo comportamientos

Integración

Paradigma reactivo

Por siempre
avanzar()
Si hay_precipicio()
detenerse()

A. WEIGEL.
MOVING TOY.
APPLICATION FILED MAR. 20, 1911.
1,017,066. Patented Feb. 13, 1912.

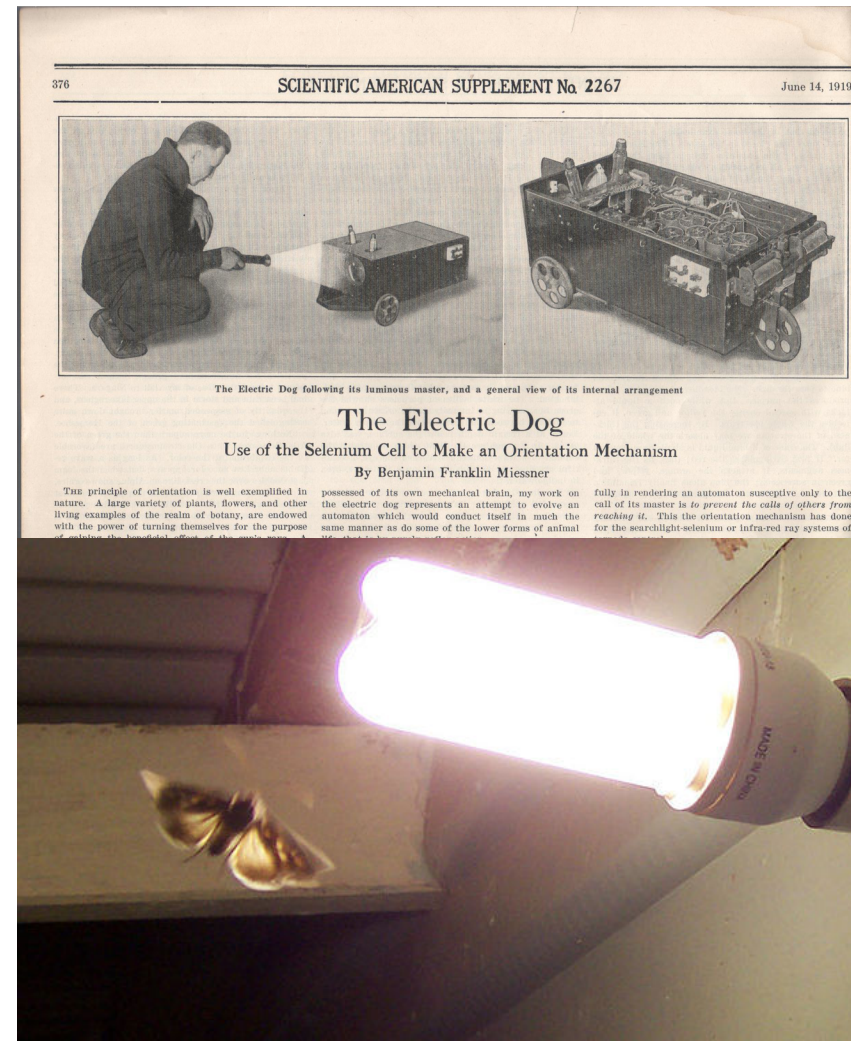


Witnesses:
Max Lemmer
Paul Gschäpfer

Inventor:
Adolf Weigel

Paradigma reactivo

```
Por siempre  
avanzar()  
Si luz_izq()>luz_der()  
    giro_izq()  
Sino  
    giro_der()
```

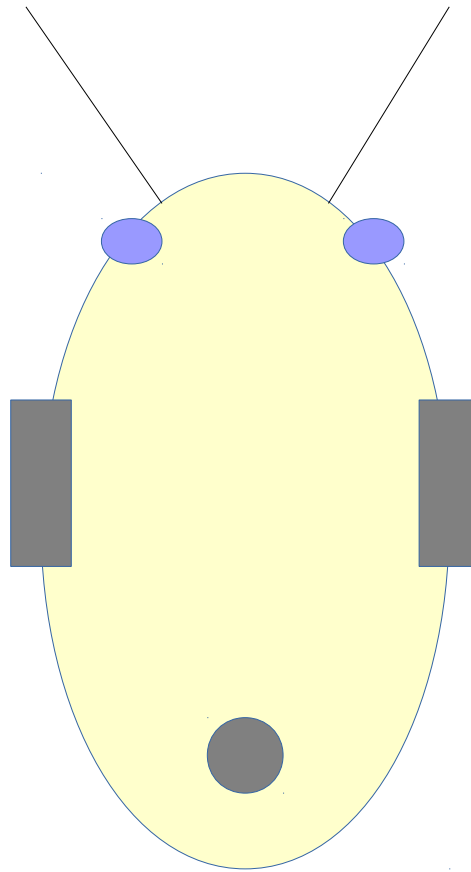


Desarrollo de un robot reactivo

Ejemplo de aplicación, basada en un insecto.

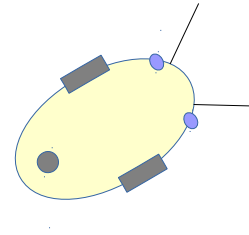
- Qué hace:
 - De noche recorre el piso recogiendo migas.
 - Huye de la luz, pegándose a las paredes.
 - Cuando huye, al llegar a un rincón espera la oscuridad y que se le pase el susto.
- Descripción del robot y del ambiente:
 - Se mueve en una habitación.
 - control diferencial (2 ruedas)
 - 2 sensores de luz (ojos) y 2 de contacto (antenas)

Desarrollo de un robot reactivo



Desarrollo de un robot reactivo

De noche recorre el piso recogiendo migas.



Por siempre
avanzar()

Si paso_tiempo() o antena_izq() o antena_der()
cambio_dirección()

Desarrollo de un robot reactivo

Huye de la luz, pegándose a las paredes.



Por siempre

avanzar()

Si luz_izq()>luz_der()

giro_der()

Sino

giro_izq()

Por siempre

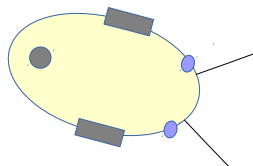
avanzar()

Si antena_izq()

giro_der()

Si antena_der()

giro_izq()

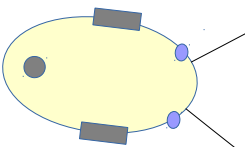


Desarrollo de un robot reactivo

Cuando huye, al llegar a un rincón espera la oscuridad y que se le pase el susto.

```
Si antena_izq() y antena_der()  
    detener()
```

```
mientras luz_izq() o luz_der() o paso_tiempo()  
    espero()
```



Desarrollo de un robot reactivo

Cuando huye, al llegar a un rincón espera la oscuridad y que se le pase el susto.

```
Si antena_izq() y antena_der()  
    detener()
```

```
mientras luz_izq() o luz_der() o paso_tiempo()  
    espero()
```

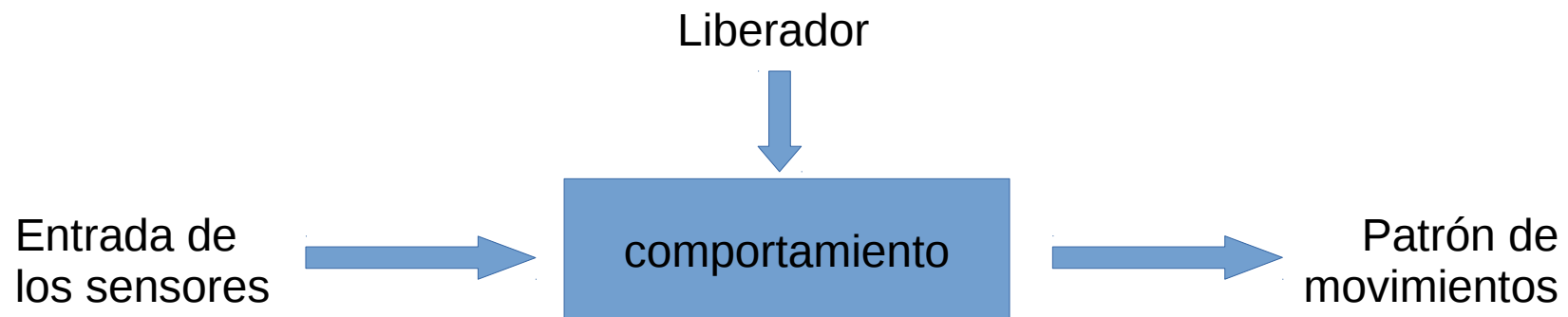
Combinando comportamientos

- Los comportamientos corren en paralelo, y pueden producir salida en simultáneo
 - Si las salidas son compatibles, no hay problema.
 - Si deciden cosas distintas, hay que elegir: prioridad.
- Un comportamiento puede inhibir a otro:
Si estoy escapando, no intento comer.
- Un comportamiento puede activar a otro:
Si encontré un escondite, espero.

Combinando comportamientos

Innate Release Mechanism (von Frisch, Lorenz & Tinbergen, 1973):
cada comportamiento tiene un *liberador*.

- El liberador actúa como una señal de control para activar un comportamiento.
- Si un comportamiento no se libera, no responde a los estímulos sensoriales y no produce salida motora.



Combinando comportamientos

Huye de la luz, pegándose a las paredes.

```
Por siempre  
  avanzar()
```

```
  Si antena_izq()  
    giro_der()  
  Si antena_der()  
    giro_izq()
```

```
Por siempre  
  avanzar()
```

```
  Si luz_izq()>luz_der()  
    giro_der()  
  Sino  
    giro_izq()
```

```
Por siempre  
  avanzar()
```

```
  Si antena_izq()  
    giro_der()  
  Sino si antena_der()  
    giro_izq()
```

```
  Sino si luz_izq()>luz_der()  
    giro_der()  
  Sino  
    giro_izq()
```

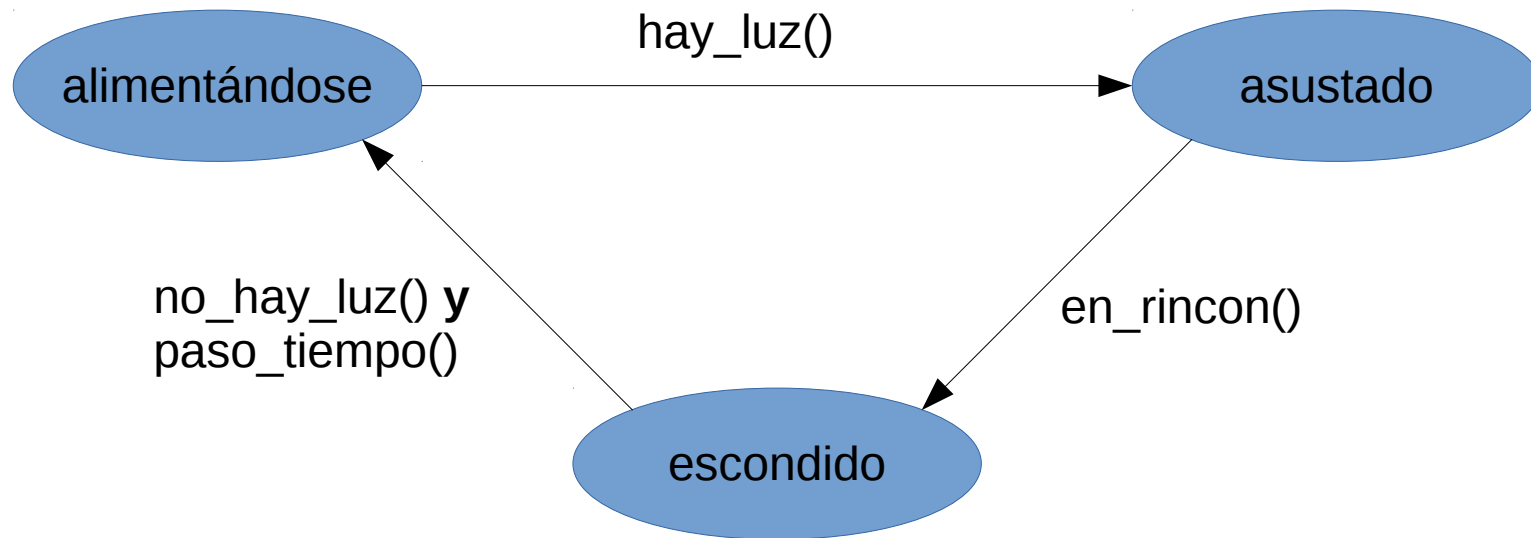
¡El orden importa!

Combinando comportamientos

- Los comportamientos solo viven en el presente: solo sentido y acción, no tienen memoria.
- Hay veces que mantener estado es inevitable: una percepción en el pasado puede condicionar un comportamiento actual. En el ejemplo:
 - Cuando se prende la luz empieza a huir y esconderse, sin importar si la luz se apaga: está *asustado*.
- El estado determina la relación entrada – salida: En estado diferentes respondemos de manera diferente a las mismas entradas.
- Cuanto menos estados, mejor.

Combinando comportamientos

Estados: {alimentándose, asustado, escondido}



Combinando comportamientos

- Cambiar de estado puede verse como un tipo de acción.
- Decidir en función de un estado puede verse como leer un sensor. A veces es posible reemplazar un estado con sensado:

```
Si estado=='alimentandose'  
y encuentre_miga()  
  levantar_miga()  
  estado = 'cargando'  
...  
Si estado=='cargando'  
  //volver al nido
```

```
Si estado=='alimentandose'  
y encuentre_miga()  
  levantar_miga()  
...  
Si estado=='alimentandose'  
y siento_carga()  
  //volver al nido
```

Desarrollando un robot reactivo

Cuando huye, al llegar a un rincón espera la oscuridad y que se le pase el susto.

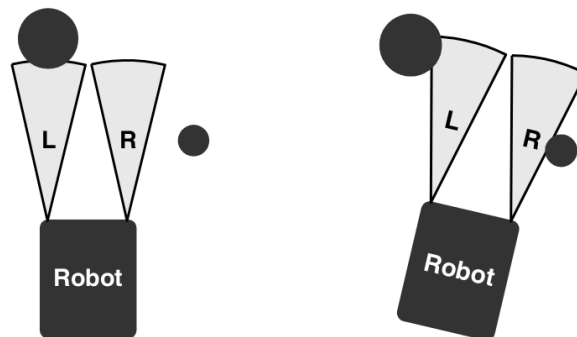
```
Si estado=='asustado'  
  Si antena_izq() y antena_der()  
    estado='escondido'  
  Sino si antena_izq()  
    giro_der()  
  Sino si antena_der()  
    giro_izq()  
  Sino si luz_izq()>luz_der()  
    giro_der()  
  Sino  
    giro_izq()
```

```
Si estado=='comiendo'  
  Si hay_luz()  
    estado='asustado'  
  Si paso_tiempo()  
    o antena_izq() o antena_der()  
      cambio_dirección()
```

```
Si estado=='escondido'  
  Si paso_tiempo()  
  y no hay_luz()  
    estado='comiendo'
```

Paradigma híbrido

- A veces es ventajoso planificar.
 - El conocimiento del entorno tiene que ser una ventaja.
- A veces es inevitable planificar.
 - Los mínimos locales pueden ser letales.

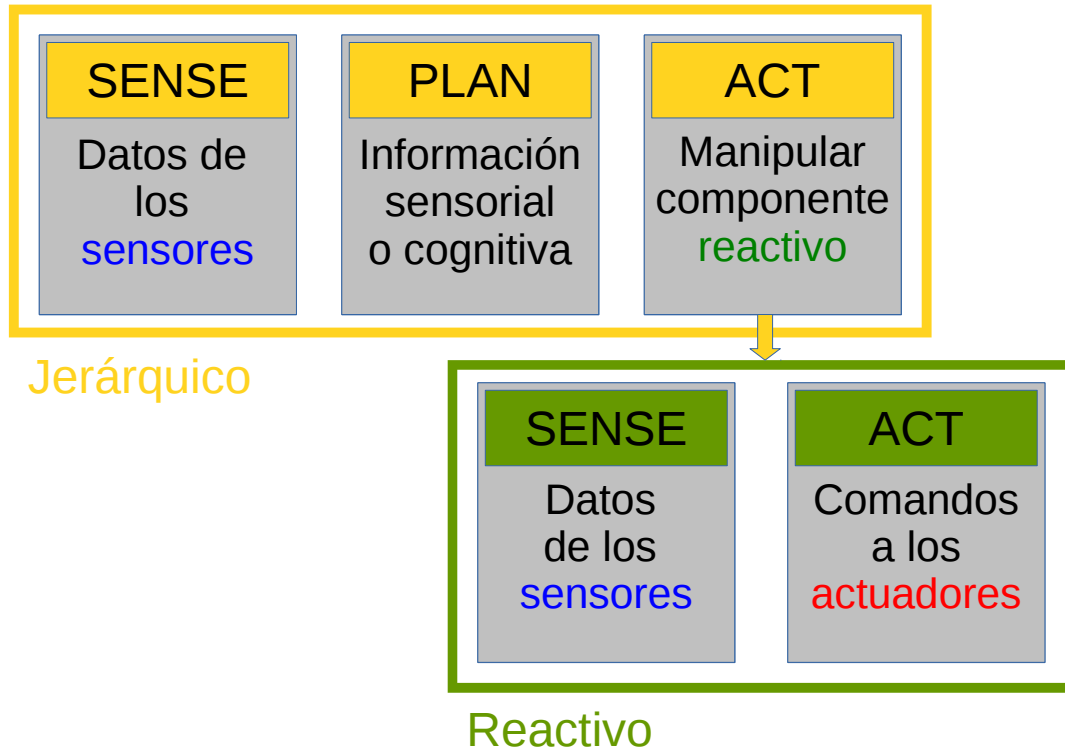


Paradigma híbrido

Ejemplo: robot de carga en un depósito, guarda y recoge cajas de estanterías.

- Jerárquico
 - Planificar orden de trabajo.
 - Coordinar con otros robots.
 - Planificar camino óptimo.
- Reactivo
 - Seguir pasillos.
 - Evitar colisiones.
 - Apoyar y levantar cajas
- ¿Cómo interaccionan?

Paradigma híbrido



- El sistema reactivo controla las acciones de tiempo real del robot y lo mantiene fuera de peligro.
- El sistema jerárquico se ejecuta de forma asincrónica, y le configura al sistema reactivo los objetivos que debe seguir.

Fin.

Referencias

- Cybernetic Zoo
<http://cyberneticzoo.com/>
- S. Russell, P. Norvig, Artificial Intelligence: a Modern Approach, Prentice Hall – 2010
- R. Murphy, Introduction to AI Robotics, MIT Press, 2000.
- Polémica en Scientific American, Enero 1990
- K. Ford, P. Hayes, On Computational Wings: Rethinking the Goals of Artificial Intelligence. Exploring Intelligence, SciAm, 1998