Agentes Inteligentes

Capítulo 2

Intelijencia

Intelijencia, dame
el nombre exacto de las cosas!
. . . Que mi palabra sea
la cosa misma
creada por mi alma nuevamente.
Que por mí vayan todos
los que no las conocen, a las cosas;
que por mí vayan todos
los que ya las olvidan, a las cosas;
que por mí vayan todos
los mismos que las aman, a las cosas. . .
¡Intelijencia, dame
el nombre exacto, y tuyo,
y suyo, y mío, de las cosas!

Juan Ramón Jiménez.

Outline

♢ Los agentes y sus ambientes.

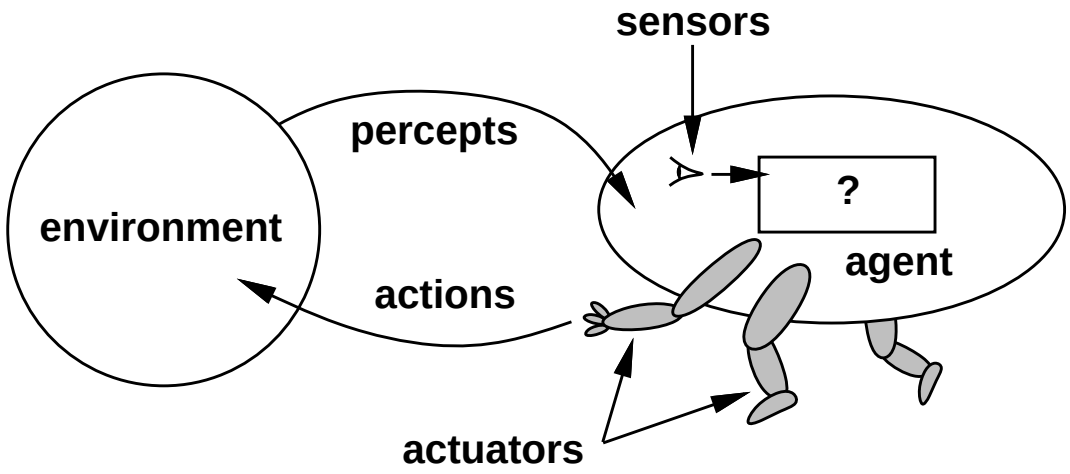
♢ Racionalidad

♢ Medidas de rendimiento, el ambiente, los actuadores y los sensores, PEAS (Performance measure, Environment, Actuators, Sensors)

♢ Tipos de ambientes

♢ Tipos de agentes

Los agentes y sus ambientes



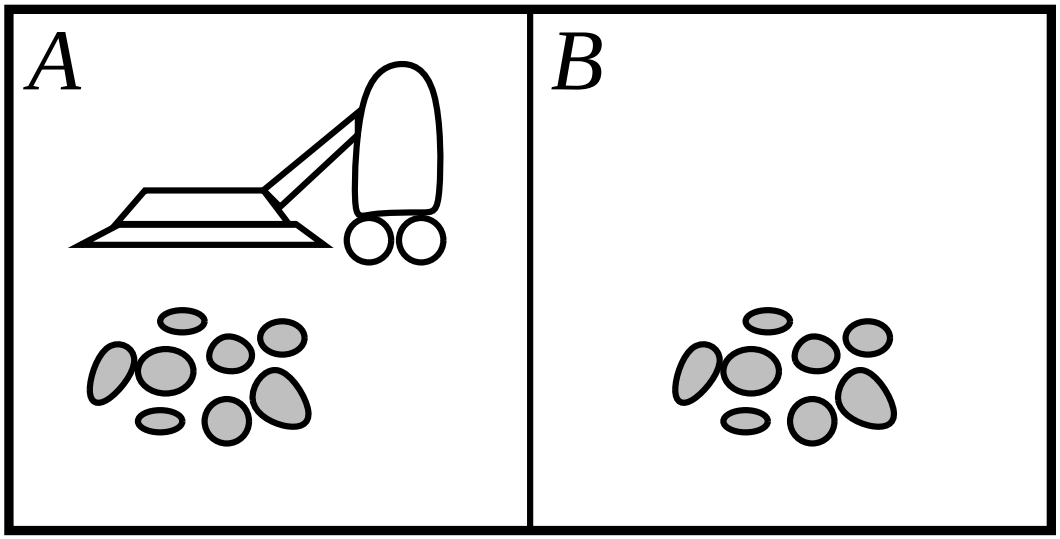
Los Agentes pueden ser humanos, robots, softbots, dispositivos como el termostato y muchos otros.

La función agente transforma historias de "perceptos" en acciones:

$$f : \mathcal{P}^* \rightarrow \mathcal{A}$$

El programa agente se ejecuta sobre una arquitectura física particular para "producir" f

El mundo de la aspiradora



Perceptos: ubicación y contenidos, e.g., $[A, Dirty]$

Acciones: $left$, $right$, $suck$, $noOp$

Un agente aspiradora

Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean]$, $[A, Clean]$	$Right$
$[A, Clean]$, $[A, Dirty]$	$Suck$
⋮	⋮

function REFLEX-VACUUM-AGENT([*location*,*status*]) **returns** an action

if $status = Dirty$ **then return** $Suck$
else if $location = A$ **then return** $Right$
else if $location = B$ **then return** $Left$

¿Qué es la función Right?
¿Podemos implementarla con un pequeño programa agente?

Racionalidad

Una medida fija de rendimiento evalua una secuencia de ambientes

– ¿un punto por cuadro limpio en un tiempo T?

– ¿un punto por cuadro limpio por unidad de tiempo, menos uno por movimiento?

– ¿Penalización por $> k$ cuadro sucios?

Un agente racional escoge aquella acción que maximiza el valor esperado de su medida de rendimiento respecto a cierta secuencia de perceptos

Racional $\neq$ omnisciente
Racional $\neq$ clarividente
Racional $\neq$ exitoso

Racional $\Rightarrow$ exploración, aprendizaje, autonomía.

PEAS

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

Considere, por ejemplo, el diseño de un taxi automático:

Medidas de rendimiento??

Ambiente??

Actuadores??

Sensores??

PEAS

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

Considere, por ejemplo, el diseño de un taxi automático:

Medidas de rendimiento?? seguridad, destinos posibles, ganancias, status legal, confort, . . .

Ambiente?? calles y avenidas, tráfico, peatones, clima, . . .

Actuadores?? volante, acelerador, freno, bocina, corneta, tablero, . . .

Sensores?? video, velocímetro, valvulas, sensores del motor, teclados, GPS, ...

En agente comprador en Internet

Medidas de rendimiento??

Ambiente??

Actuadores??

Sensores??

Tipos de ambientes

	Solitario	Backgammon	Compras en Internet	Taxi
Observable?? Determínistico?? Episódico?? Estático?? Discreto?? Un-solo-agente??				

Tipos de ambientes

	Solitario	Backgammon	Compras en Internet	Taxi
Observable??	Si	Si	No	No
Determínistico??				
Episódico??				
Estático??				
Discreto??				
Un-solo-agente??				

Tipos de ambiente

	Solitario	Backgammon	Compras en Internet	Taxi
Observable??	Si	Si	No	No
Determínistico??	Si	No	Parcialmente	No
Episódico??				
Estático??				
Discreto??				
Un-solo-agente??				

Tipos de ambientes

	Solitario	Backgammon	Compras en Internet	Taxi
Observable??	Si	Si	No	No
Determínistico??	Si	No	Parcialmente	No
Episódico??	No	No	No	No
Estático??				
Discreto??				
Un-solo-agente??				

Tipos de ambientes

	Solitario	Backgammon	Compras en Internet	Taxi
Observable??	Si	Si	No	No
Determínistico??	Si	No	Parcialmente	No
Episódico??	No	No	No	No
Estático??	Si	Semi	Semi	No
Discreto??				
Un-solo-agente??				

Tipos de ambientes

	Solitario	Backgammon	Compras en Internet	Taxi
Observable??	Si	Si	No	No
Determínistico??	Si	No	Parcialmente	No
Episódico??	No	No	No	No
Estático??	Si	Semi	Semi	No
Discreto??	Si	Si	Si	No
Un-solo-agente??				

Tipos de ambientes

	Solitario	Backgammon	Compras en Internet	Taxi
Observable??	Si	Si	No	No
Determínistico??	Si	No	Parcialmente	No
Episódico??	No	No	No	No
Estático??	Si	Semi	Semi	No
Discreto??	Si	Si	Si	No
Un-solo-agente??	Si	No	Si (salvo subastas)	No

El tipo de ambiente determina, en buena medida, el diseño del agente.

El mundo real es (¿desde luego?) parcialmente observable, estocástico, secuencial, dinámico, continuo y multiagente.

Tipos de agentes

Listamos 4 tipos básicos en orden de generalidad creciente (segu'n R&N):

– Agente simple reflejo

– Agente reflejo con estado

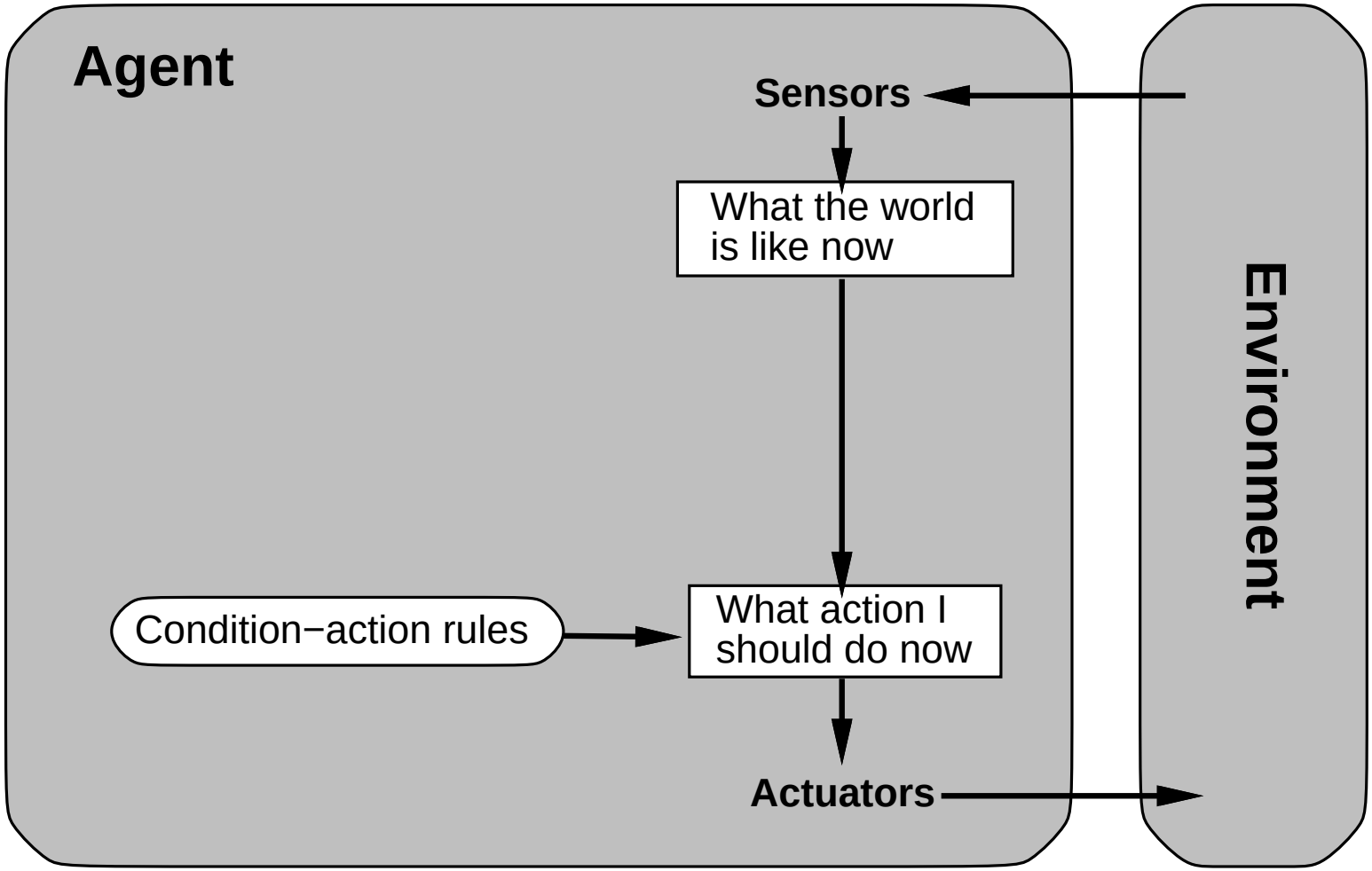
– Agentes orientado a metas

– Agentes basados en utilidad

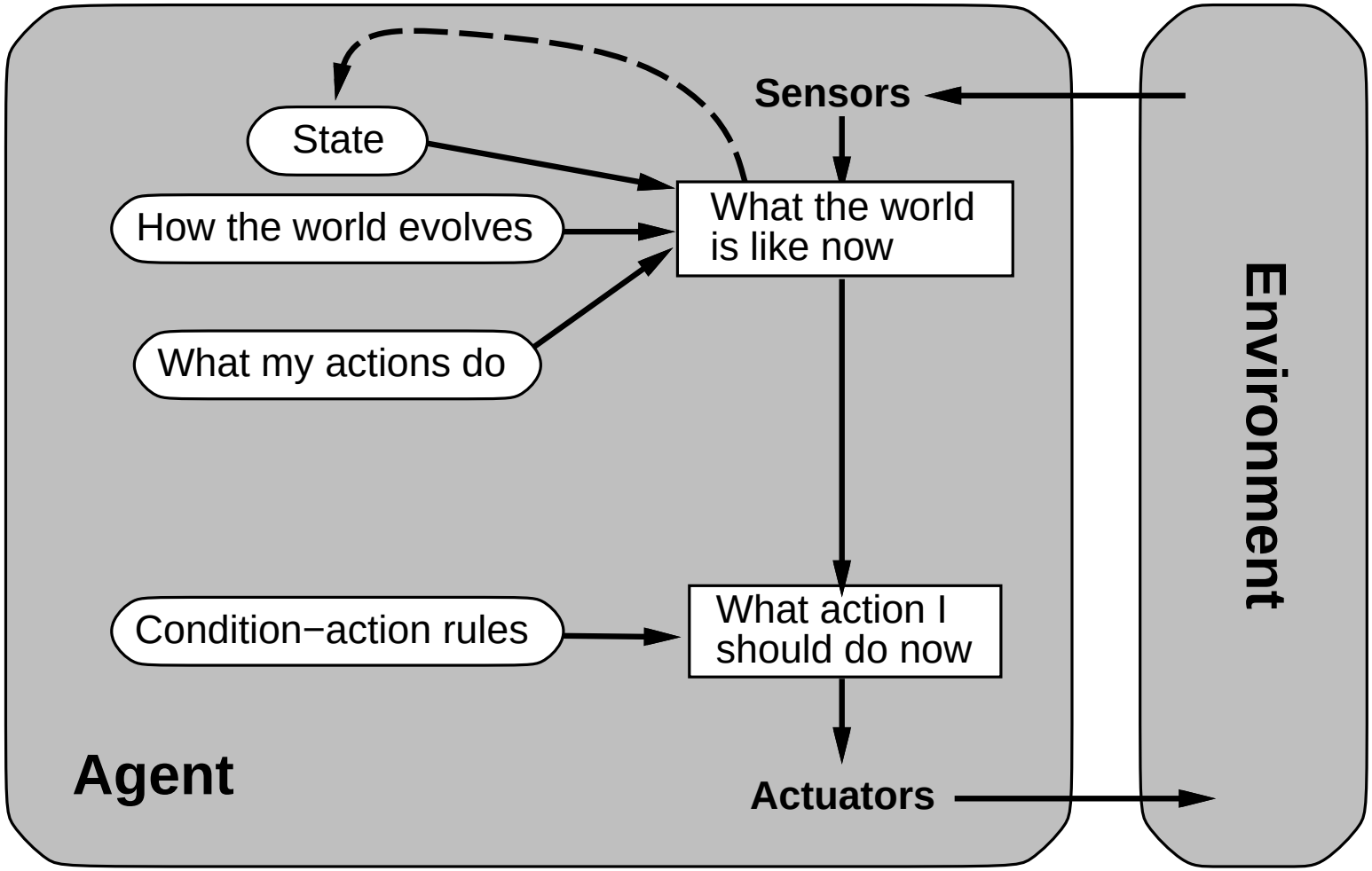
Todos se pueden convertir en agentes aprendices.

Ver también la jerárquia de [Davila, Uzcátegui, Tucci].

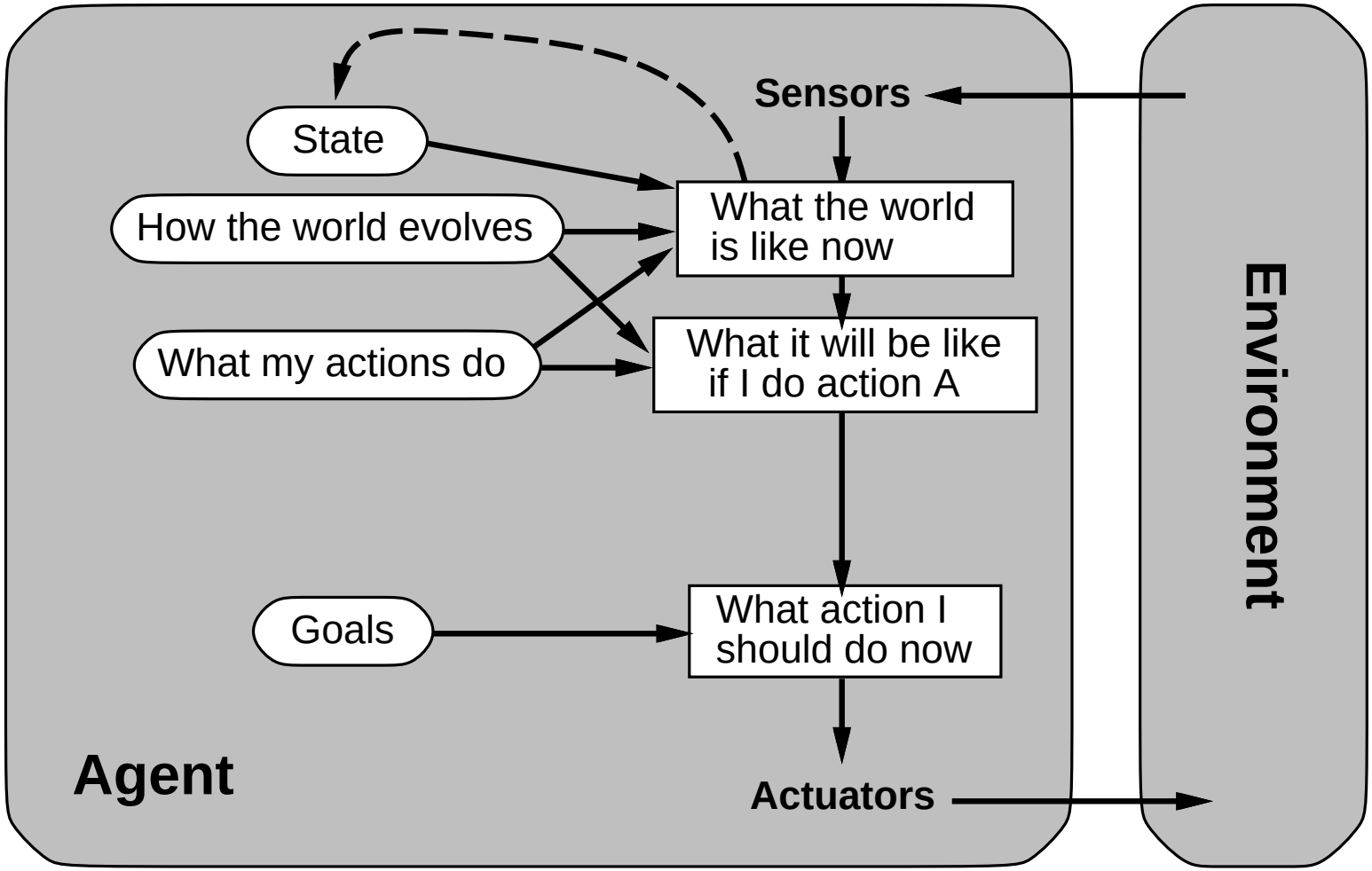
Agente simple reflejo



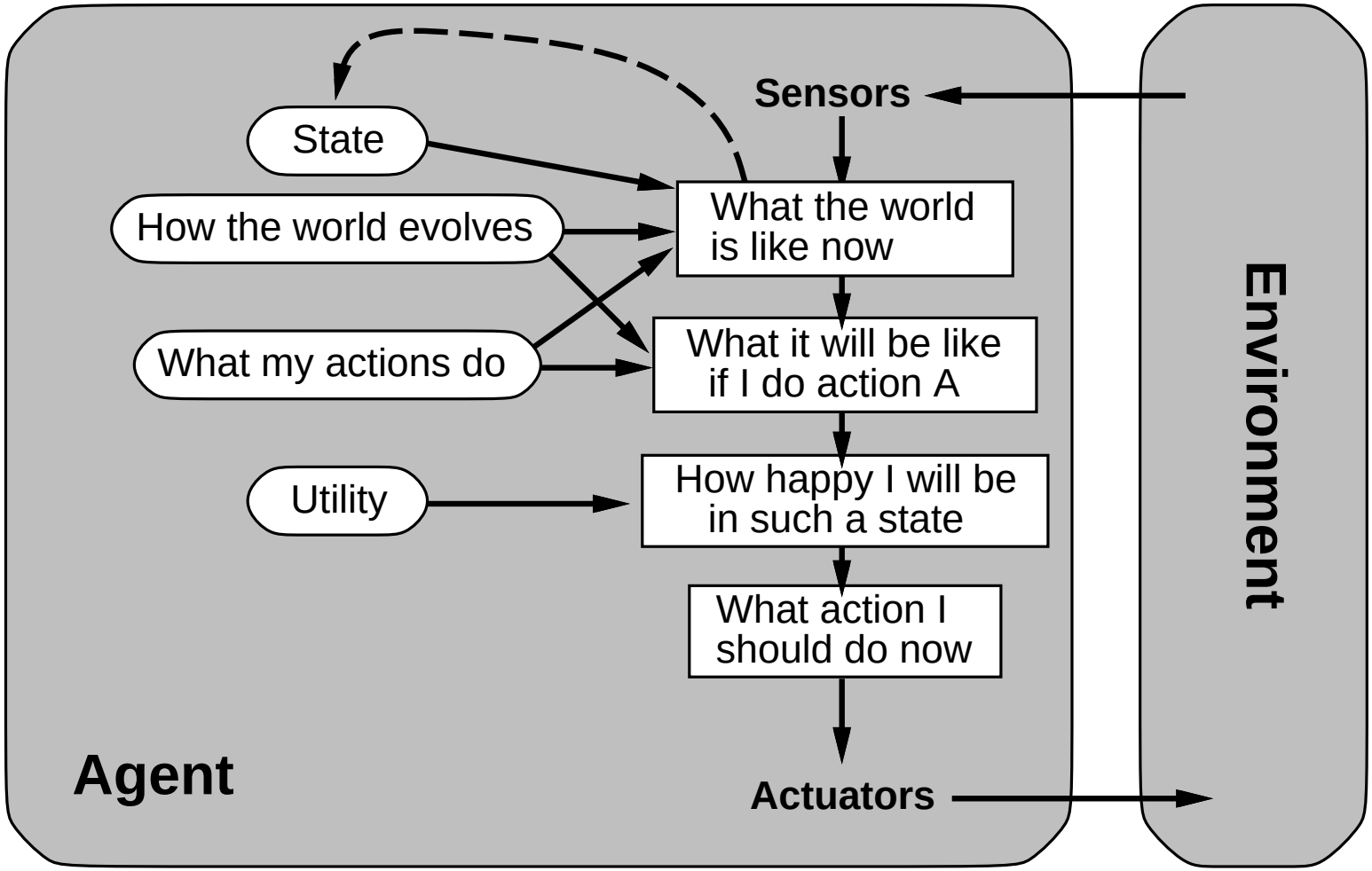
Agente reflejo con estado



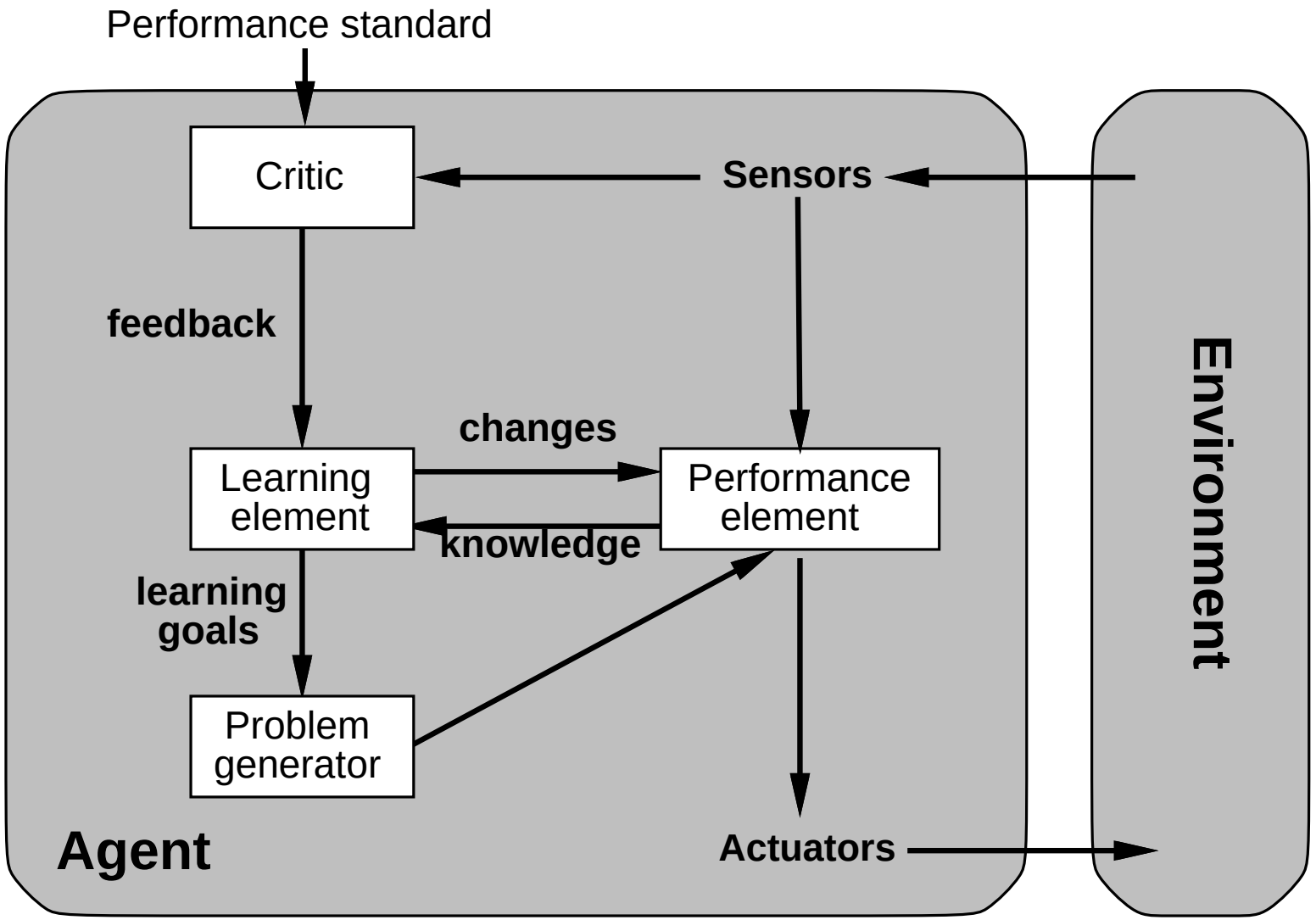
Agente orientado a metas



Agente basado en utilidad



Agente aprendiz



código AIMA

```
(setq joe (make-agent :name 'joe :body (make-agent-body)
                      :program (make-dumb-agent-program)))

(defun make-dumb-agent-program ()
  (let ((memory nil))
    #'(lambda (percept)
        (push percept memory)
        'no-op)))
```

código Bioinformantes: un tutor

```
% Integrity constraints.
if biotutor_requested, not(class_opened) then open_class. \\
if class_opened, not(seen_something) then show_page_1.\\
if class_opened, seen_page_1, not(pending_queries) then
close_class.\\
if seen_page_1, want_a_demo then do_method.\\
if user_continue, method_running then do_yes.\\
if question_asked then answer_question.\\

% Definitions
to seen_something do seen_page_1.\\
to seen_something do user_continue.
```

código Galatea: Un colono

```
executable(invade). executable(tala). executable(cultiva).
executable(invade_vecino). executable(anexa).
executable(afiliate). executable(vende_tierra).
executable(subsiste). executable(compra_tierra).
executable(siembra_ganado).

observable(tienes_tierra). observable(tienes_dinero).
observable(extension_posible).
observable(te_solicitan_apoyo_politico).

si not(tienes_tierra), not(tienes_dinero) entonces
invade_y_subsiste.\\
si tienes_tierra, extension_posible entonces invade_vecino.\\
si te_solicitan_apoyo_politico, te_dan_tierra entonces afiliate.
si tienes_tierra, not(tienes_dinero) entonces
vende_y_subsiste_con_eso.\\
si tienes_tierra, tienes_dinero entonces explotacion_ganadera.
```

```
para invade_y_subsiste haga
    invade, % define un terreno para este agente.
    tala,   % inicia una trayectoria de tala en ese terreno,
            % lineal, 20 hectareas en 5 a#os.
    cultiva.    % inicia una trayectoria de cultivo en ese terreno

para invade_vecino haga
    anexa.  % inicia una trayectoria de tala o extendiende
        % la del resto de su propiedad
        % extendiende la trayectoria del capital.

para vende_y_subsiste_con_eso haga
    vende_tierra, % Inicia trayectoria de capital con intereses.
    subsiste. % Inicia una trayectoria de gastos de supervivencia.

para explotacion_ganadera haga
    compra_tierra,  % afecta trayectoria del capital.
    siembra_ganado. % inicia trayectoria del reba#o y del capital.
```

código Galatea: Un politiquero

```
executable(solicita_apoyo). executable(ofrece_tierra).

observable(tienes_tierra). observable(observo_colono).

% Integrity constraints.
si observo_colono, tienes_tierra entonces negocia_tierra_por_voto.

% Definitions
para negocia_tierra_por_voto haga
    solicita_apoyo,
    ofrece_tierra.

observe tienes_tierra. observe observo_colono.
```