

INTELLIGENZA ARTIFICIALE

INTRODUZIONE

Riferimenti

- ◇ S. Russell, P. Norvig, "Artificial Intelligence: a Modern Approach", Prentice Hall, 2002 (II edizione)
- ◇ Lucidi (<http://www.math.unipd.it/~sperduti/intelligenza-artificiale.html>)

Contenuti del corso

- ◇ Introduzione, Motivazioni, Architetture di Agenti Intelligenti
- ◇ Risoluzione di Problemi e Cenni di Sistemi con Vincoli
- ◇ Giochi come Problemi di Ricerca
- ◇ Rappresentazione della conoscenza e ragionamento
 - calcolo proposizionale
 - calcolo dei predicati
 - deduzione automatica
 - introduzione alla programmazione logica
- ◇ Pianificazione
- ◇ Cenni di Apprendimento Automatico
- ◇ Trattamento dell'Incertezza
 - approccio probabilistico,
 - reti bayesiane

Modalità esame: progetto più esame orale (o compito)

Forme di Progetto

Lo studente può scegliere tra le seguenti forme di progetto:

- Proattivo

Su proposta dello studente, previa discussione con il docente. Sono, ad esempio, incoraggiate attività relative all'introduzione di tecniche dell'Intelligenza Artificiale all'interno di sistemi sviluppati per i corsi di Ingegneria del Software della (ex) Laurea Triennale.

- Contributivo

Alcuni degli algoritmi presentati nel libro di testo sono stati implementati e resi disponibili in un "Online Code Repository". Gli algoritmi sono implementati nei seguenti linguaggi: Python, Java, Lisp.

Diversi algoritmi non sono ancora implementati (in alcun linguaggio, o per alcuni linguaggi, ad esempio Java) e possono quindi costituire oggetto del progetto.

Forme di Progetto

- Conoscitivo/Comparativo

Lo studente "sperimenta" vari sistemi esistenti e reperibili in rete e li compara con spirito critico rispetto a varie dimensioni, fra cui: facilità d'uso, facilità di comprensione/modificabilità del codice sorgente, prestazioni rispetto al dominio applicativo.

La relazione finale del progetto assume in questo caso ruolo preminente.

- Esplorativo

Lo studente elabora e sperimenta idee proprie, come ad esempio euristiche relative ad un particolare dominio applicativo.

Progetto

Gruppi di lavoro:

- Sono ammessi gruppi di lavoro non superiori a 3 studenti (preferibilmente 2 studenti).

Prodotti del progetto:

- Relazione finale (in HTML) che descrive l'attività svolta, incluso almeno un caso di studio [BONUS: 2 punti in più nel voto finale se si presenta il progetto tramite presentazione pubblica!]
- Codice prodotto (nei casi in cui è previsto)
- Consuntivo orario delle singole attività svolte.

Cosa trattiamo oggi ?

- ◇ La nostra definizione di agente intelligente
- ◇ Introduzione al concetto di ambiente
 - esempio molto semplice di ambiente
- ◇ Il concetto di Razionalità,
 - misura di prestazione,
 - ambiente operativo,
 - azioni,
 - sensori
- ◇ Proprietà degli ambienti → complessità computazionale
- ◇ Possibili tipologie di agente → capacità computazionali diverse

Agente Intelligente (o razionale)

Un **agente** è una entità che percepisce ed agisce

Nel contesto del corso, un **agente razionale**, cerca di raggiungere il più possibile i suoi obiettivi data l'informazione disponibile, ovvero cerca di massimizzare il soddisfacimento dei propri bisogni sfruttando tutte le informazioni di cui dispone o che può acquisire con le sue azioni

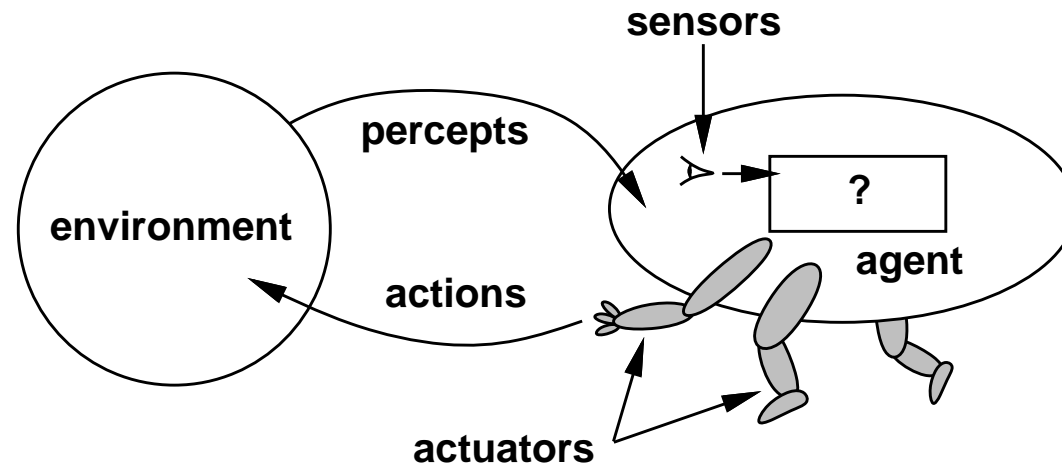
Da un punto di vista astratto, un agente è una funzione da tutte le possibili sequenze di percezioni \mathcal{P}^* ad azioni (ammissibili) \mathcal{A} :

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

Fra tutte le classi di ambienti e compiti, cerchiamo l'agente (o classe di agenti) con le prestazioni migliori

Caveat: *limitazioni computazionali impediscono la realizzazione di razionalità perfetta* → progettare il miglior **programma** date le risorse rese disponibili dalla macchina

Agenti e ambienti



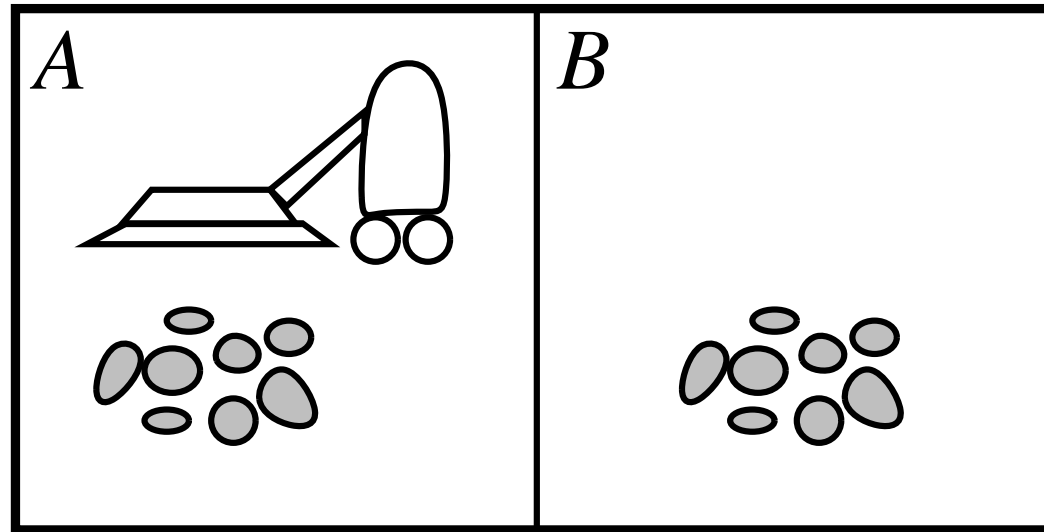
Agenti: includono umani, robot, softbot, attuatori, etc.

La **funzione agente** mappa da tutte le possibili sequenze di percezioni ad azioni \mathcal{A} :

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

Il **programma agente** “gira” sulla **architettura fisica** per produrre f

Il mondo dell'aspirapolvere !



Percezioni: locazioni e contenuti, e.g., [*A*, *Dirty*]

Azioni: *Left*, *Right*, *Suck*, *NoOp*

Un agente aspirapolvere !

Percept sequence	Action
<i>[A, Clean]</i>	<i>Right</i>
<i>[A, Dirty]</i>	<i>Suck</i>
<i>[B, Clean]</i>	<i>Left</i>
<i>[B, Dirty]</i>	<i>Suck</i>
<i>[A, Clean], [A, Clean]</i>	<i>Right</i>
<i>[A, Clean], [A, Dirty]</i>	<i>Suck</i>
⋮	⋮

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*

Quale è la giusta funzione ?

Può essere implementata in un semplice e piccolo programma agente ?

Razionalità

Viene fissata una **misura di prestazione** che valuta la **sequenza di percezioni**

- +1 per ogni spazio pulito in tempo T ?
- +1 per ogni spazio pulito per istante di tempo, -1 per spostamento ?
- ...

Un **agente razionale** sceglie una qualunque azione che massimizza il valore **aspettato** della misura di prestazione **data la sequenza di percezioni ottenuta fino all'istante corrente**

Razionalità \neq Omniscienza

Razionalità \neq Chiaroveggenza

Razionalità \neq Successo

Razionalità \Rightarrow esplorazione, apprendimento, autonomia

PEAS (Performance Environment Actuators Sensors)

Per progettare un agente razionale dobbiamo specificare l' **ambiente operativo**, all'interno del quale si deve svolgere il compito

Consideriamo, ad esempio, il compito di progettare un taxi automatizzato:

Misura di Prestazioni??

Ambiente operativo??

Attuatori??

Sensori??

PEAS (Performance Environment Actuators Sensors)

Per progettare un agente razionale dobbiamo specificare l' **ambiente operativo**, all'interno del quale si deve svolgere il compito

Consideriamo, ad esempio, il compito di progettare un taxi automatizzato:

Misura di Prestazioni?? sicurezza, destinazione, profitto, comodità, ...

Ambiente operativo?? strade e autostrade, traffico, pedoni, tempo, ...

Attuatori?? volante, acceleratore, freni, clacson, microfono/video, ...

Sensori?? telecamera, accelerometri, sensori del motore, GPS, ...

Agente che acquista su Internet

Misura di Prestazioni??

Ambiente operativo??

Attuatori??

Sensori??

Tipi di ambienti

	Solitario	Backgammon	Internet shopping	Taxi
<u>Osservabile??</u>				
<u>Deterministico??</u>				
<u>Episodico??</u>				
<u>Statico??</u>				
<u>Discreto??</u>				
<u>Agente singolo??</u>				

Tipi di ambienti

	Solitario	Backgammon	Internet shopping	Taxi
<u>Osservabile??</u>	Si	Si	No	No
<u>Deterministico??</u>				
<u>Episodico??</u>				
<u>Statico??</u>				
<u>Discreto??</u>				
<u>Agente singolo??</u>				

Tipi di ambienti

	Solitario	Backgammon	Internet shopping	Taxi
<u>Osservabile??</u>	Si	Si	No	No
<u>Deterministico??</u>	Si	No	In parte	No
<u>Episodico??</u>				
<u>Statico??</u>				
<u>Discreto??</u>				
<u>Agente singolo??</u>				

Tipi di ambienti

	Solitario	Backgammon	Internet shopping	Taxi
<u>Osservabile??</u>	Si	Si	No	No
<u>Deterministico??</u>	Si	No	In parte	No
<u>Episodico??</u>	No	No	No	No
<u>Statico??</u>				
<u>Discreto??</u>				
<u>Agente singolo??</u>				

Tipi di ambienti

	Solitario	Backgammon	Internet shopping	Taxi
<u>Osservabile??</u>	Si	Si	No	No
<u>Deterministico??</u>	Si	No	In parte	No
<u>Episodico??</u>	No	No	No	No
<u>Statico??</u>	Si	Semi	Semi	No
<u>Discreto??</u>				
<u>Agente singolo??</u>				

Tipi di ambienti

	Solitario	Backgammon	Internet shopping	Taxi
<u>Osservabile??</u>	Si	Si	No	No
<u>Deterministico??</u>	Si	No	In parte	No
<u>Episodico??</u>	No	No	No	No
<u>Statico??</u>	Si	Semi	Semi	No
<u>Discreto??</u>	Si	Si	Si	No
<u>Agente singolo??</u>				

Tipi di ambienti

	Solitario	Backgammon	Internet shopping	Taxi
<u>Osservabile??</u>	Si	Si	No	No
<u>Deterministico??</u>	Si	No	In parte	No
<u>Episodico??</u>	No	No	No	No
<u>Statico??</u>	Si	Semi	Semi	No
<u>Discreto??</u>	Si	Si	Si	No
<u>Agente singolo??</u>	Si	No	Si (eccetto aste)	No

Il tipo di ambiente determina largamente la progettazione dell'agente

Il mondo reale è, ovviamente, parzialmente osservabile, stocastico, sequenziale, dinamico, continuo, multi-agente

Tipi di agenti

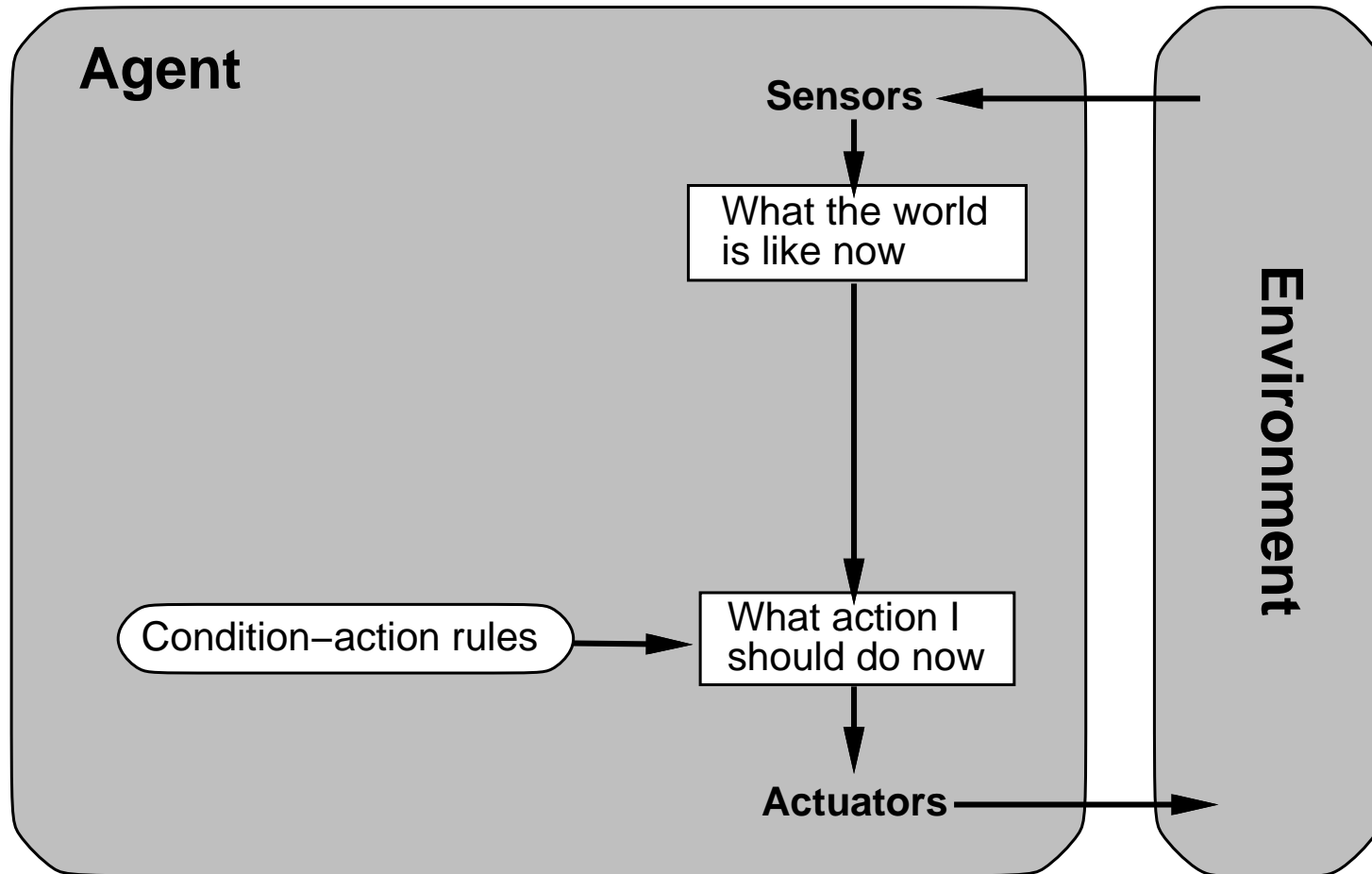
agente = architettura + programma

Si possono definire in generale quattro tipi di agente:

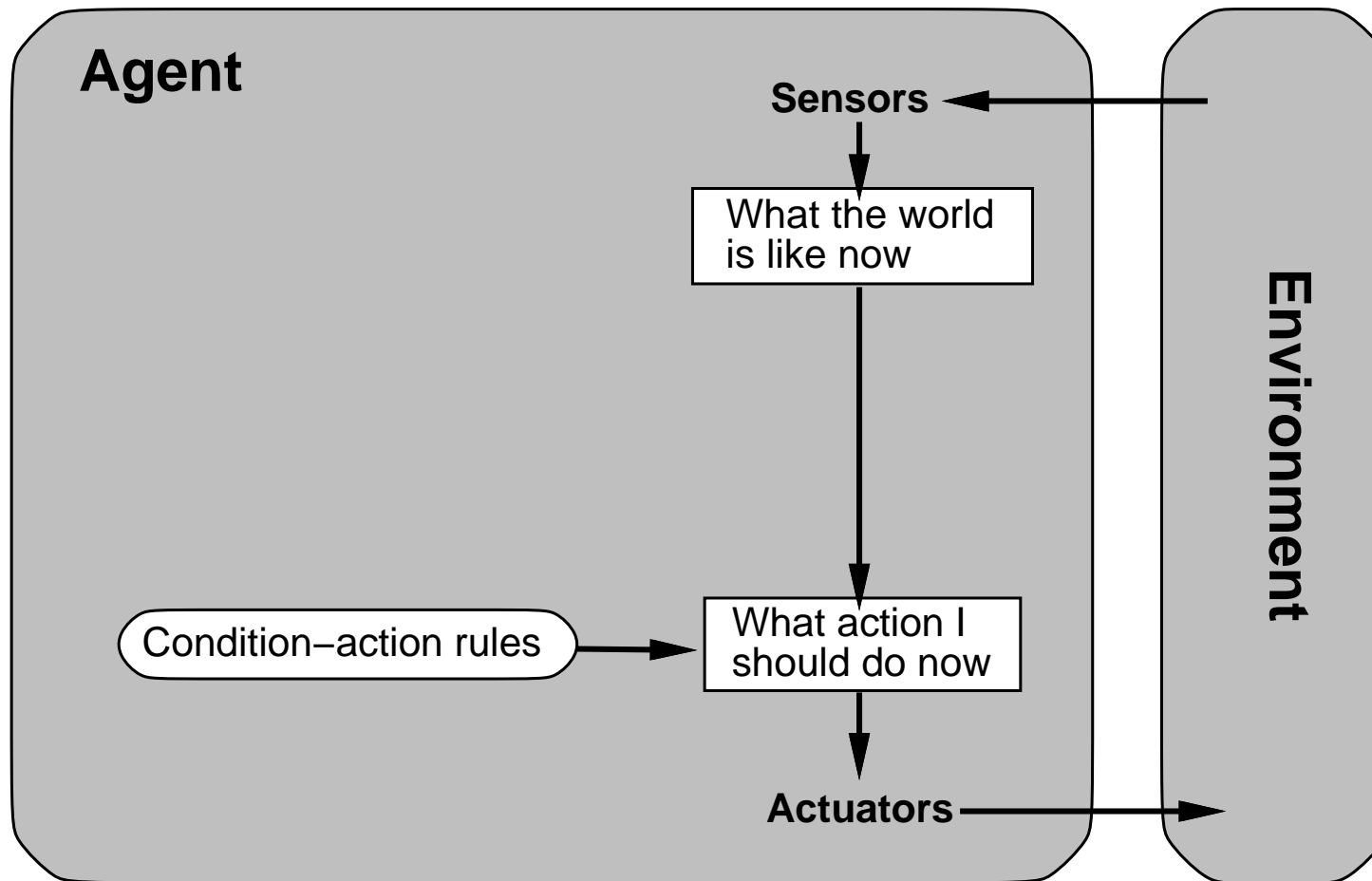
- agenti a riflesso semplice
- agenti a riflesso con stato
- agenti basati su goal
- agenti basati su una misura di utilità

Tutti questi tipi di agenti possono essere trasformati in agenti che apprendono

Agenti a riflesso semplice

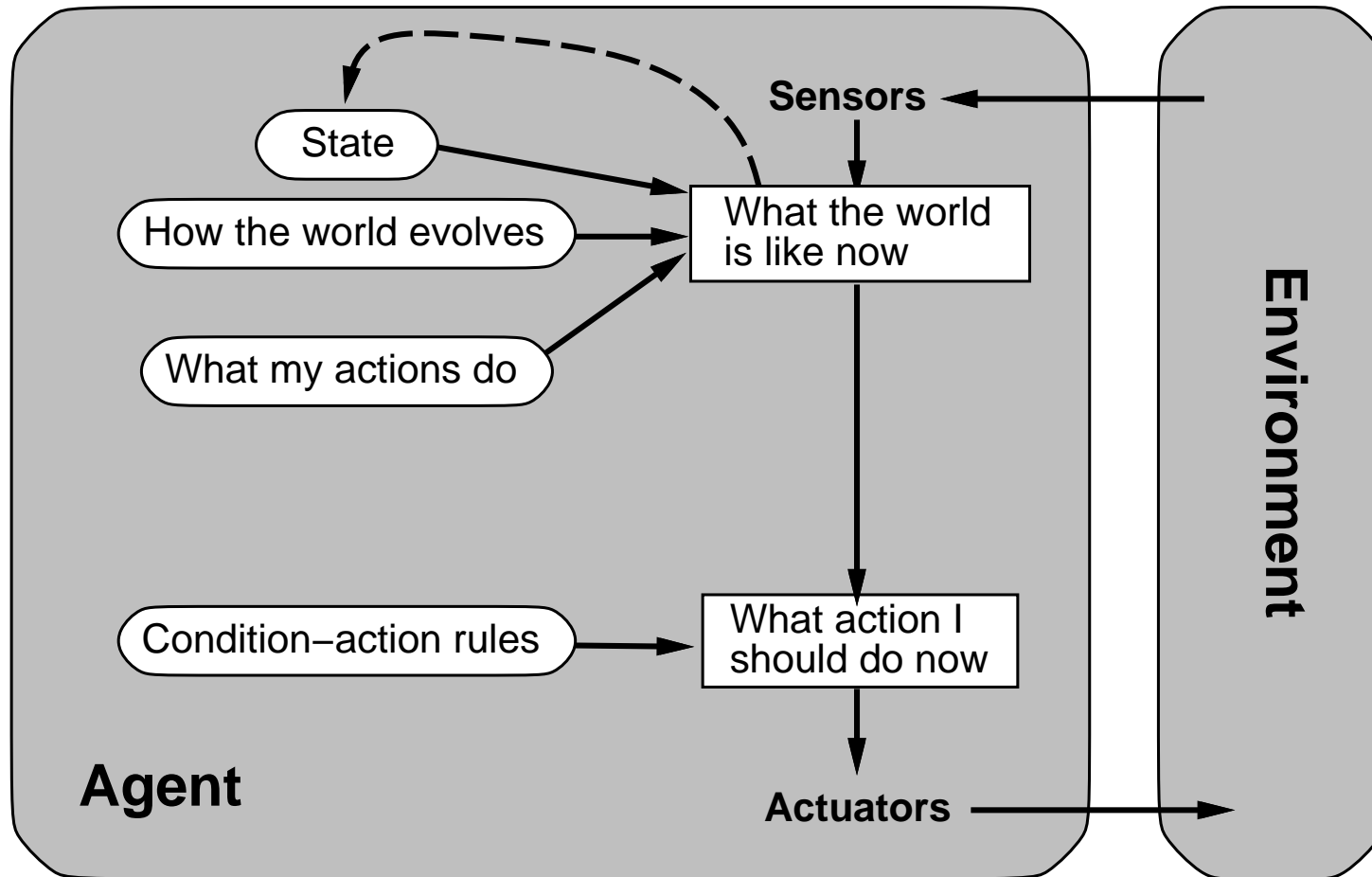


Agenti a riflesso semplice

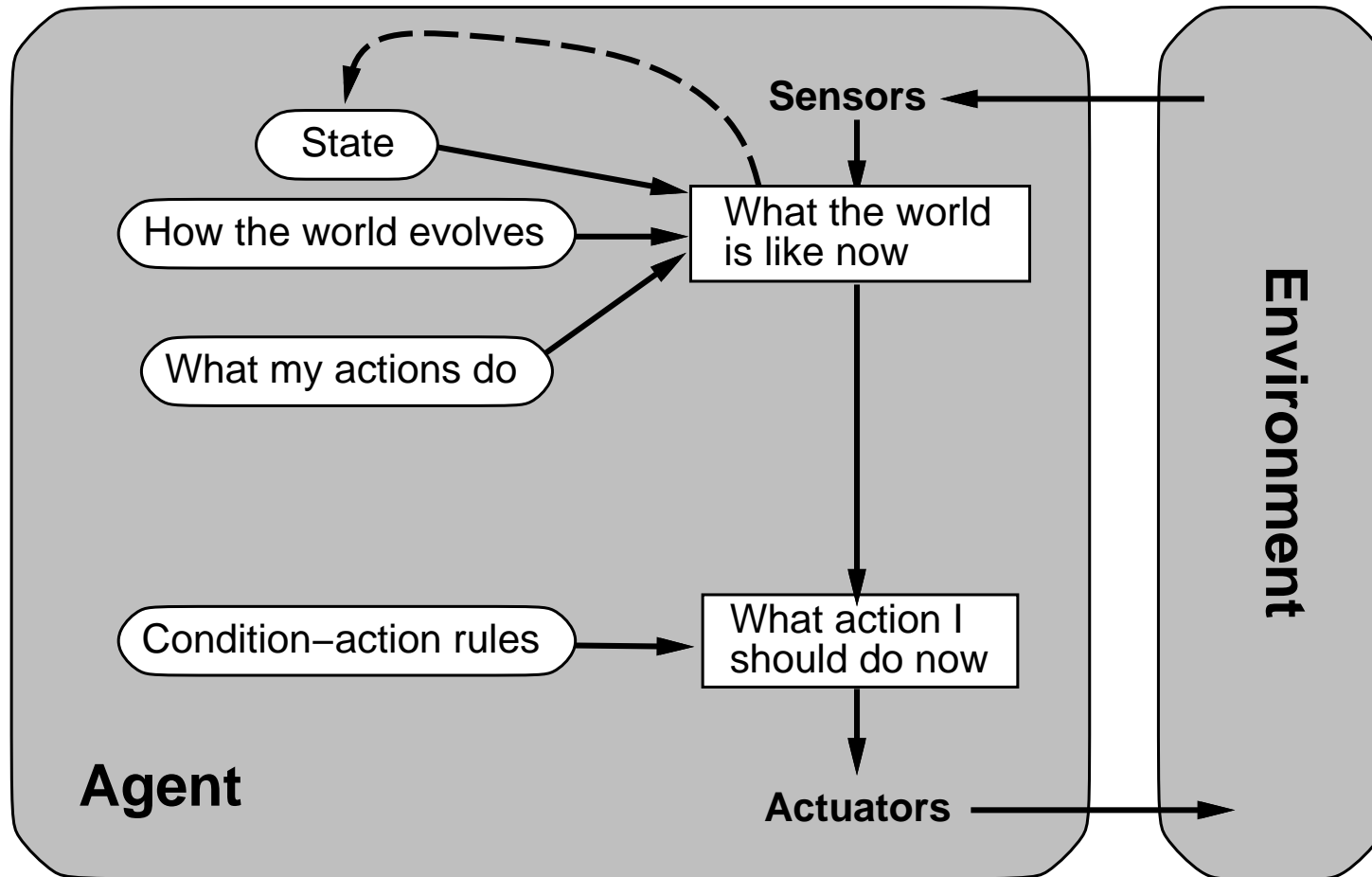


Problema: se l'ambiente è parzialmente osservabile può fallire ! (randomizzare)
Soluzione migliore: tenere traccia delle percezioni passate → stato interno

Agenti a riflesso con stato

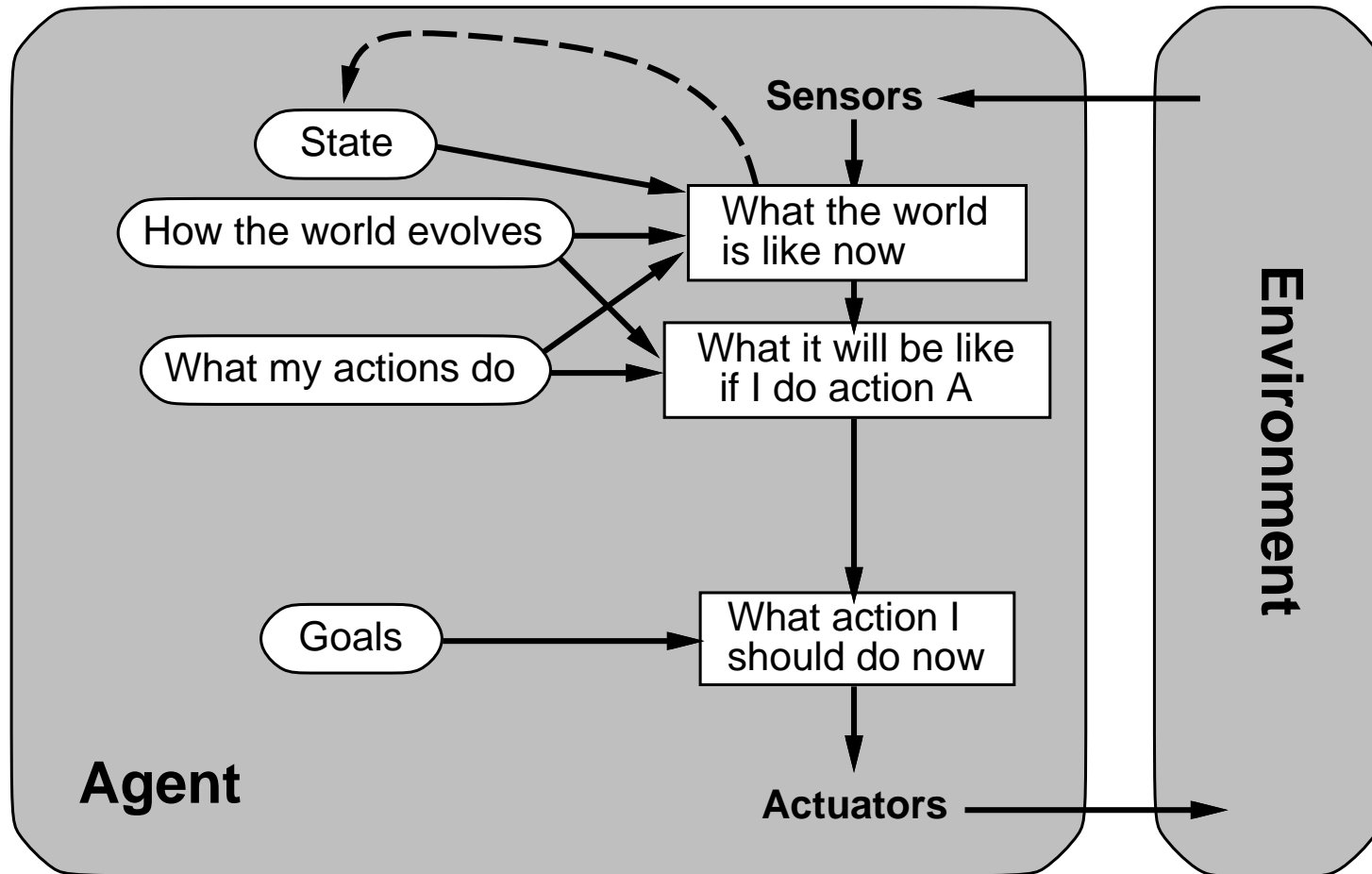


Agenti a riflesso con stato

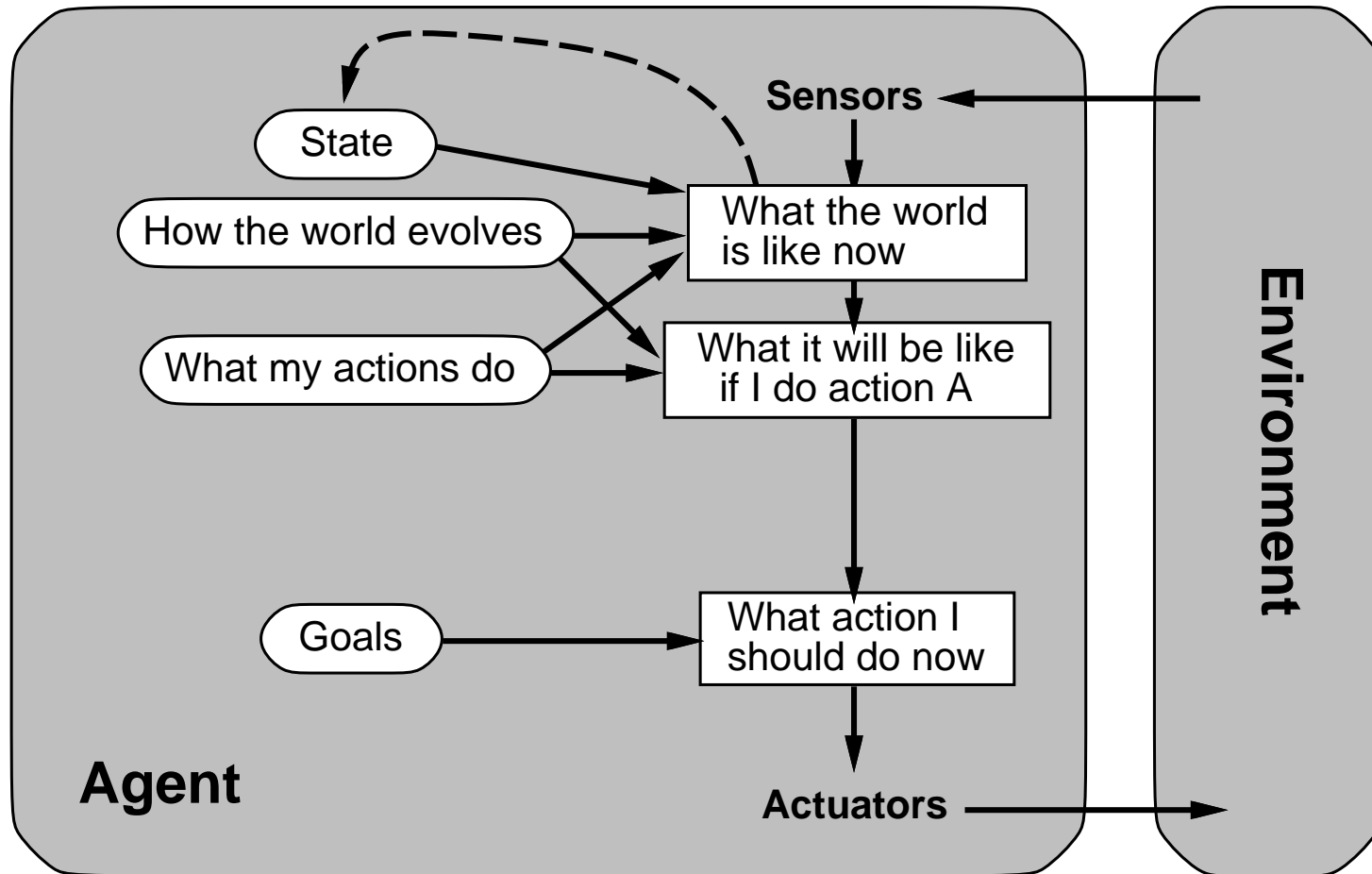


Problema: poco flessibile perché il comportamento è codificato direttamente nelle regole! Soluzione: introduzione di goal

Agenti basati su goal

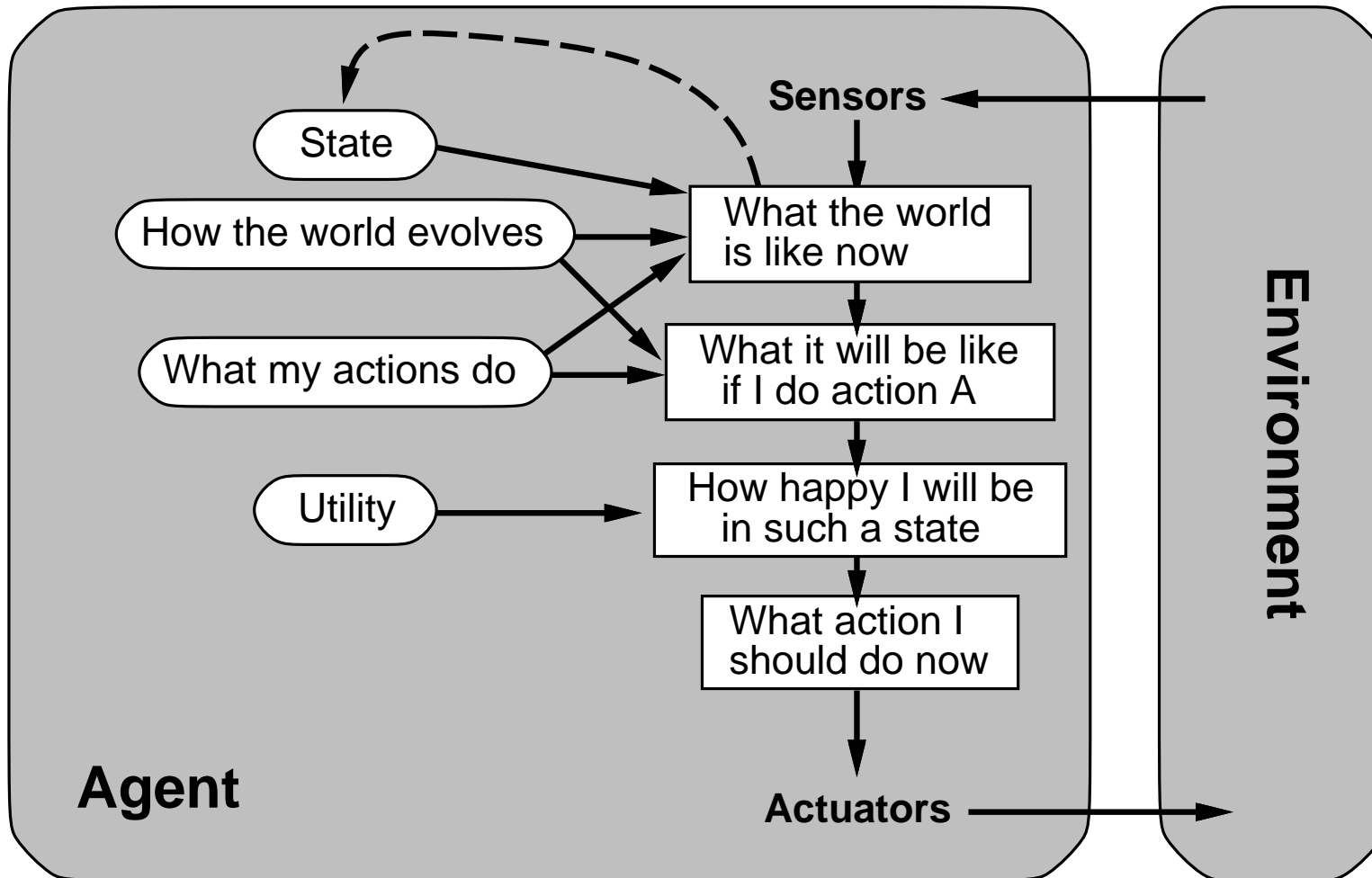


Agenti basati su goal



Problema: se ho più goal in conflitto fra loro o che non riesco a raggiungere con certezza ? Soluzione: misura di utilità

Agenti basati su una misura di utilità



Agenti che apprendono

