

Implementazione di un Sistema di Supporto Mnemonico

Giorgio Ravera

A.A 2007/2008

Indice

I	Introduzione all'Intelligenza Artificiale	1
1	Agenti Intelligenti	2
1.1	Definizione di Agente	2
1.2	Ambienti	3
1.2.1	Tipologie di Ambienti	4
1.3	Struttura di un agente	5
1.3.1	Agenti Reattivi semplici	6
1.3.2	Agenti basati su Modello	7
1.3.3	Agenti basati su Obiettivi	7
1.3.4	Agenti basati sull'Utilità	7
1.3.5	Agenti che Apprendono	7
1.3.6	Agenti basati sulla Conoscenza	7
2	Comunicazione	8
3	Rappresentazione della Conoscenza	9
4	Apprendimento	10
	Bibliografia	11

Parte I

Introduzione all'Intelligenza Artificiale

Capitolo 1

Agenti Intelligenti

1.1 Definizione di Agente

Un **agente** è un entità che agisce, cioè svolge un'azione. Deve essere in grado di percepire l'ambiente che lo circonda attraverso dei **sensori** ed eseguire delle azioni attraverso degli **attuatori**.

Un **agente intelligente** o **razionale** è un agente che agisce in modo da ottenere il migliore risultato o, in caso di incertezza, il migliore risultato atteso.

Consideriamo un esempio comune di agente: una persona umana. Quando un soggetto si trova a dover risolvere un problema per prima cosa analizza l'ambiente esterno, attraverso i (occhi e orecchie). In secondo luogo prova ad elaborare, nella sua mente, un'idea del problema e prova ad elaborare una soluzione ottimale correlando tra loro le informazioni in suo possesso (pregresse e acquisite). Infine, attraverso gli attuatori (braccia, piedi o, più in generale, muscoli) la applica.

A questa descrizione molto elementare possiamo ricondurre ogni singola attività, dalla soluzione di un problema matematico, alla scelta di quale canale televisivo guardare oppure semplicemente a quale strada percorrere quando si è davanti a un bivio.

Un agente è composto da due elementi fondamentali ed indivisibili:

- **architettura**: il sistema fisico (corpo umano, hardware di un calcolatore) che compone l'agente e gli consente di eseguire i calcoli.
- **programma**: un insieme di istruzioni che guidano l'agente nella risoluzione dei problemi e consentano di interpretare i segnali provenienti dal mondo esterno e ricodificare i risultati per controllare gli attuatori.

Esistono diverse tipologie di agenti che si differenziano in base alla struttura che implementano. Per ogni tipologia vi è un approccio teorico specifico.

Nei paragrafi successivi verranno approfonditi aspetti legati all'ambiente esterno e alla struttura di un agente.

1.2 Ambienti

Un agente interagisce continuamente con l'ambiente che lo circonda:

- preleva informazioni attraverso i sensori
- compie azioni attraverso gli attuatori

Con il termine **percezione** si indicano gli input che l'agente ottiene, tramite i sensori, dall'ambiente esterno in un dato istante. La **sequenza percettiva** è la storia completa di tutto ciò che l'agente ha percepito nella sua esistenza.

In generale la scelta dell'azione di un agente in un qualsiasi istante può dipendere dall'intera sequenza percettiva osservata fino a quel momento. Se possiamo specificare l'azione prescelta dall'agente per ogni possibile sequenza percettiva, allora abbiamo descritto l'agente in modo completo. Ciò implicherebbe avere una conoscenza completa dell'ambiente esterno e questo non è sempre possibile, come vedremo in seguito.

Si identifica con **funzione agente** l'insieme di azioni che descrivono il comportamento dell'agente in risposta a ciascun elemento della sequenza percettiva. La si può vedere come una tabella che mette in corrispondenza di ogni percezione una delle possibili azioni. Nel caso non si conoscesse tale funzione è possibile ricavarla (interamente o solo parte di essa) provando tutte le possibili sequenze percettive e registrando il comportamento dell'agente.

La tabella appena descritta ha una validità *esterna* all'agente. Al suo interno ci sarà un **programma agente** che implementerà tale funzione. E' importante sottolineare la differenza: la funzione è una descrizione matematica astrazione, il programma è una sua implementazione concreta in esecuzione sull'architettura dell'agente.

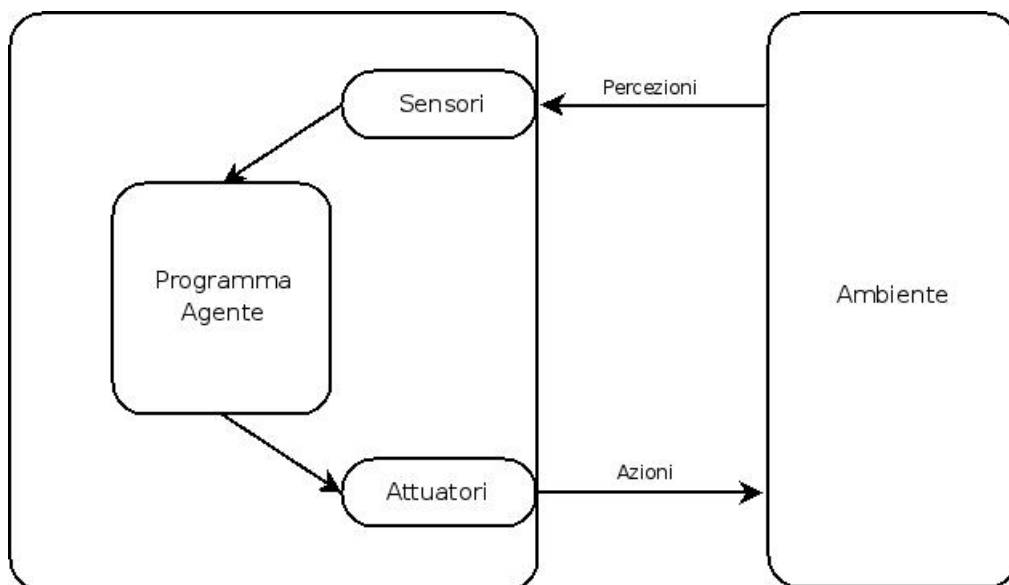


Figura 1.1: Schema di un generico Agente

La figura 1.1 illustra visivamente quanto detto fin ora: come si può vedere viene separato dall'ambiente esterno l'agente che è composto da sensori, attuatori e programma agente. Non è presentato un dettaglio di quest'ultimo per mantenere il più generica possibile la presentazione. Nel paragrafo 1.3 ne verranno mostrate le diverse implementazioni.

1.2.1 Tipologie di Ambienti

Per poter realizzare un agente con un dato obiettivo, è necessario definire l'ambiente che lo circonda. Da esso, infatti, l'agente trarrà tutte le informazioni necessarie per poter effettuare le decisioni e gli elementi per poterle attuare.

E', quindi, necessario identificare delle proprietà in base a cui suddividerli in categorie. Queste proprietà determinano in gran parte la progettazione di agenti appropriati e l'applicabilità delle principali tecniche alla loro implementazione.

Le principali proprietà sono le seguenti:

- **osservabilità completa/parziale:** caratterizza il grado d'accesso dei sensori al mondo esterno. Un ambiente parzialmente osservabile può essere dovuto a sensori inaccurati o imprecisi, alla presenza di rumore oppure all'impossibilità di ottenere alcune informazioni.
- **deterministico/stocastico:** nel caso di determinismo l'agente riesce a prevedere lo stato successivo dell'ambiente sulla base della misura fatta e dell'azione. Qualora ciò non fosse possibile si parla di ambiente stocastico. Nel caso in cui l'ambiente è deterministico ad eccezione delle azioni che altri agenti potrebbero compiere si parla di ambiente **strategico**.
- **episodico/sequenziale:** un ambiente si dice episodico se ogni azione generata non dipende dalle precedenti. Sequenziale nel caso in cui più percezioni influenzano la scelta dell'azione da svolgere.
- **statico/dinamico:** un ambiente è statico nel caso in cui non può cambiare nel periodo in cui l'agente ragiona al fine di prendere la decisione ottima. E' dinamico quando può variare durante la fase di ragionamento. Ciò comporta grossi problemi perché l'azione generata potrebbe non essere più quella corretta. Se l'ambiente non cambia con il tempo ma la valutazione dell'agente si allora si dice **semidinamico**.
- **discreto/continuo:** tale classificazione è applicata allo *stato* dell'ambiente e a come viene rappresentato il *tempo* alle percezioni e azioni dell'agente. Può essere discreto, campionato ad intervalli regolari, oppure continuo, analizzato con continuità senza salti.
- **agente singolo/multiagente:** si parla di ambiente a singolo agente nel caso in cui eventuali altri elementi autonomi nell'ambiente non influenzano le scelte effettuate. Si parla di multiagente quando altri agenti possono partecipare attivamente al raggiungimento dell'obiettivo.

Si può capire facilmente che il caso più complesso è dato dalla combinazione di ambiente: *parzialmente osservabile, stocastico, sequenziale, dinamico, continuo e multiagente*.

1.3 Struttura di un agente

Come già visto in precedenza vale la seguente legge:

$$agente = architettura + programma \quad (1.1)$$

Con **architettura** si identifica la componente fisica, materiale dell'agente, ovvero l'hardware, i circuiti e le periferiche che lo compongono. E' composta prevalentemente da quattro differenti elementi:

- **strumenti di calcolo:** componenti che si occupano di elaborare i dati per relazionarli ed ottenere gli strumenti per poter effettuare le decisioni. In genere sono rappresentati dal/dai processore/i.
- **sensori:** analizzano l'ambiente esterno e codificano le informazioni raccolte in dati analizzabili dagli apparati di calcolo. Sono in genere microfoni, tastiere, videocamere o misuratori di temperatura.
- **attuatori:** componenti che ricevono istruzioni dagli apparati di calcolo e li traducono in azioni, come ad esempio il movimento di un braccio o la produzione di un segnale visivo/uditivo. Alcuni esempi possono essere display o casse acustiche.
- **memoria:** in alcuni agenti è presente la possibilità di memorizzare informazioni per poter avere più dati da tenere in considerazione nell'elaborazione di decisioni. Possono essere realizzati mediante database memorizzati su dischi o nastri.

Il **programma agente** è un software che deve essere in grado di:

- acquisire correttamente i dati provenienti dai sensori
- codificare tali dati per ottimizzare la funzione di decisione/ragionamento
- inviare agli attuatori azioni da svolgere compatibili con la loro progettazione

Come si può osservare dalla sua definizione, tale programma deve essere progettato per una specifica architettura e, quindi, risultare compatibile con i sensori e gli attuatori scelti.

E' importante sottolineare la differenza tra il programma agente che prende come input solamente la percezione corrente e la funzione agente il cui input è costituito dall'intera storia delle percezioni. Il programma agente si basa sulla sola percezione corrente perché l'ambiente non può fornirgli nulla di più. Se le sue azioni dipendono dalla sequenza percettiva precedente è necessario utilizzare un sistema di memorizzazione.

Il seguente pseudocodice dovrebbe aiutare a capire la struttura base di un programma:

```
class agente-semplce
{
    map<perception, action> m;

    action calcola(perceptions p)
    {
        return m.lookup(p);
    }
}
```

Esistono sei tipologie diverse di agenti associate a una opportuna tipologia di programma agente che rappresentano i principi alla base di quasi tutti i sistemi intelligenti:

- agenti reattivi semplici
- agenti basati su modello
- agenti basati su obiettivi
- agenti basati sull'utilità
- agenti che apprendono
- agenti basati sulla conoscenza

1.3.1 Agenti Reattivi semplici

Un *agente reattivo semplice* è un agente che sceglie le azioni sulla base della sola percezione corrente, ignorando completamente quelle passate. In altre parole non ha memoria.

Un tale agente è, appunto, molto semplice da realizzare ma ha una intelligenza molto limitata. Si può schematizzare il suo operato attraverso il seguente pseudocodice:

```
class agente-reattivo-semplce
{
    map<state, list<rule>> r;
    map<rule, action> m;

    action calcola(perceptions p)
    {
        state s = code_percption(p);
        rule r1 = (r.lookup(s, p)).getFirst();
        return m.lookup(r1);
    }
}
```


Per prima cosa, viene codificata in forma astratta la percezione attraverso una sua elaborazione (funzione `code_perception`). Da qui si potrà ottenere uno stato dell'ambiente esterno. A questo punto è necessario interpretare lo stato e capire che **regola condizione-azione** attribuire a ciò che si è osservato. Nel caso in cui ci fossero più regole per lo stesso stato verrà considerata sempre e solo la prima. A questo punto si procede cercando l'azione da compiere.

Interpretiamo con un esempio quanto appena descritto. Consideriamo un sistema di guida autonomo: ci troviamo dietro ad un'automobile e dobbiamo capire quando questa frena per fare la stessa cosa. I sensori sono una serie di telecamere montate sul parabrezza, gli attuatori sono i freni. Ad intervalli regolari viene catturata un'immagine: dall'immagine (percezione) si studiano le luci di segnalazione e in particolare se sono accese o spente (stato). In base al valore appena elaborato si cerca una corrispondenza tra questo e l'azione da svolgere. Si estrarrà, in caso di luci accese una regola condizione azione del tipo:

if (**light_on**) then **start_to_break**

Con questa regola diventa semplice capire che azione svolgere, nel caso corrente frenare.

Anche gli esseri umani fanno uso di connessioni analoghe, alcune basate sull'apprendimento, altre sui riflessi. Questi aspetti verranno approfonditi nei capitoli successivi.

Un tale agente funziona solo e unicamente nel caso in cui l'ambiente sia **completamente osservabile**. Anche una minima parte di inosservabilità potrebbe causare errori molto gravi. Nel caso in esempio la struttura potrebbe non prevedere la possibilità che le luci di segnalazione della macchina davanti non funzionino, oppure non differenziarli dalle luci di posizione. Limitandosi, quindi, alla sola analisi delle luci il sistema non funziona.

Infine un tale agente tende ad eseguire **cicli infiniti** se l'azione da compiere è sempre la stessa per ogni circostanza e lo stato non può mutare per peculiarità proprie dell'azione stessa. Supponiamo di avere un robot che deve muoversi su un terreno in relazione al colore delle piastrelle: se le piastrelle sono bianche si muove a destra o in basso, se sono nere a sinistra o in alto. A un tratto arriva in una piastrella nera con alla sua sinistra un muro. L'azione che dovrebbe essere eseguita sarà il movimento verso sinistra (prima regola) ma non sarà possibile e alla percezione successiva lo stato sarà analogo al precedente.

Per ovviare a questo problema è possibile scegliere in maniera casuale le azioni da svolgere. Ciò dovrebbe evitare cicli infiniti.

1.3.2 Agenti basati su Modello

1.3.3 Agenti basati su Obiettivi

1.3.4 Agenti basati sull'Utilità

1.3.5 Agenti che Apprendono

1.3.6 Agenti basati sulla Conoscenza

Capitolo 2

Comunicazione

La **comunicazione** consiste nello scambio intenzionale di informazioni attraverso la produzione e percezione di **segni** appartenenti a un sistema convenzionale e condiviso. E' uno strumento attraverso il quale è possibile condividere informazioni.

Il **linguaggio** è il complesso sistema di messaggi strutturati che permettono a due o più entità di comunicare con chiarezza una parte della loro conoscenza. Solitamente un linguaggio prevede regole sintattiche e semantiche che consentono a due o più interlocutori di capirsi..

Capitolo 3

Rappresentazione della Conoscenza

Capitolo 4

Apprendimento

Bibliografia

- [1] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prantice Hall, 2003.

Indice analitico

agente, 2
agente intelligente, 2
agente razionale, 2
agente reattivo semplice, 6
ambienti, 3
attuatore, 2
attuatori, 5

funzione agente, 3

percezione, 3
programma agente, 3

sensore, 2
sensori, 5
sequenza percettiva, 3