

# Pianificazione e Robotica

Stefano Nolfi  
Istituto di Psicologia, CNR  
Viale Marx, 15, 00137, Roma.  
e-mail: nolfi@ip.rm.cnr.it

## Obiettivo

Lo scopo di questo capitolo è introdurre il lettore al problema del controllo di un sistema intelligente (sia esso un sistema software che "gira" all'interno del computer oppure un robot fisico che interagisce con un ambiente esterno). Studiando questo capitolo il lettore dovrebbe:

- Capire le implicazioni di diversi modi di organizzare l'architettura complessiva di un sistema intelligente.
- Farsi una idea delle problematiche che emergono nello sviluppare un sistema in grado di esibire un comportamento.
- Capire come sfruttando l'interazione tra le componenti interne del sistema e tra il sistema e l'ambiente esterno sia possibile sviluppare sistemi semplici in grado di risolvere problemi relativamente complessi.

## Introduzione

La robotica si è affermata all'interno dell'IA cosiddetta "classica" nei primi anni settanta. In tale approccio il controllo veniva visto come un problema a se che si poteva studiare ed implementare in modo relativamente indipendente dal resto del sistema e si assumeva che il robot dovesse possedere una rappresentazione completa ed esplicita del mondo esterno. A partire dagli anni ottanta, soprattutto a seguito delle ricerche condotte da Rodney Brooks al MIT, si sono sviluppati approcci diversi, che da una parte hanno integrato il sistema di controllo con il sistema percettivo e motorio e dall'altro hanno mostrato come un robot possa operare nel mondo reale anche mediante rappresentazioni incomplete ed implicite. In questo capitolo discuteremo alcune delle principali linee di ricerca a partire dagli esordi della robotica in IA.

### 1. Il Planning

Nella visione cosiddetta "classica" dell'IA di norma si assume che un sistema intelligente sia costituito da 3 componenti o macro-sistemi: (1) un sistema percettivo in grado di estrarre informazione dall'ambiente esterno; (2) un pianificatore in grado di generare una sequenza di azioni (anche detta piano) che, tenuto conto della situazione corrente, permetta al sistema di raggiungere i suoi scopi; (3) un sistema motorio o esecutore che traduca il piano in azioni effettive. Il pianificatore costituisce dunque il cuore del sistema.

Il pianificatore è un sistema che riceve in ingresso dal sistema percettivo (o direttamente dallo sperimentatore) una descrizione dello stato attuale e una descrizione dello stato che si vuole raggiungere (scopo) e, basandosi su un insieme di conoscenze, genera un piano (cioè la descrizione di una sequenza di azioni) che una volta eseguito è in grado di permettere al sistema di raggiungere lo scopo o gli scopi desiderati. Il pianificatore opera di norma su informazioni di tipo simbolico. In altre parole, lo stato del mondo, le conoscenze del

pianificatore, lo scopo e il piano sono rappresentate con una serie di espressioni simboliche appartenenti ad un linguaggio con una sintassi ben definita.

La pianificazione (planning) dunque è un metodo alternativo al problem solving (cfr. CAP. 2), e se ne differenzia per il fatto che: (1) gli stati, gli scopi e le azioni vengono rappresentati esplicitamente e (2) durante la creazione di un piano, il pianificatore può aggiungere azioni in qualsiasi punto senza necessariamente procedere in modo incrementale a partire dallo stato iniziale o dallo stato finale.

L'approccio standard al planning nasce allo Stanford Research Institute con lo sviluppo di STRIPS. L'idea chiave di questo sistema è quella di utilizzare un linguaggio ristretto e un algoritmo di ricerca specifico (adatto al dominio e al linguaggio scelto) in grado di assicurare una pianificazione efficiente piuttosto che un linguaggio generale (come ad es. un formalismo logico) e un risolutore generale di problemi per la ricerca della soluzione.

STRIPS è stato sviluppato originariamente per controllare Shakey, un robot mobile sviluppato allo Stanford Research Institute in quegli stessi anni, che doveva essere in grado di svolgere semplici compiti all'interno di un'area formata da una serie di stanze e un corridoio (ad es. spostare un oggetto da una stanza all'altra).

In STRIPS stati, obiettivi e piani vengono rappresentati con una sequenza di predicati. Supponiamo ad es. che nello stato attuale Shakey si trovi nella stanza n.3, che l'oggetto n.1 si trovi nella stanza n. 2, e che lo scopo di Shakey sia quella di far sì che l'oggetto A sia nella stanza n. 1 e che Shakey sia nella stanza n.4. Lo stato iniziale del robot e dell'ambiente può essere rappresentato nel modo seguente (dove i puntini indicano una serie di altri predicati che descrivono in modo esaustivo l'ambiente):

Posizione(Shakey, Stanza3), Posizione(OggettoA, Stanza2), .....

Lo scopo del pianificatore può essere rappresentato nel modo seguente:

Posizione(OggettoA, Stanza1), Posizione(Shakey, Stanza4)

Le azioni che Shakey può utilizzare per creare un piano possono essere rappresentate nel modo indicato qui di seguito (dove  $\neg$  indica l'operatore logico NOT, AZIONE indica il nome dell'azione, PRECONDIZIONI indica le condizioni che debbono essere soddisfatte affinché l'azione possa essere eseguita e EFFETTI indica le conseguenze dell'azione). Per brevità abbiamo indicato solo le azioni che sono necessarie per raggiungere lo scopo descritto sopra nelle condizioni di partenza indicate:

Operazione(AZIONE: vai(x,y),  
PRECONDIZIONI: Posizione(Shakey,x),  
EFFETTI: Posizione(Shakey,y),  $\neg$ Posizione(Shakey,x))

Operazione(AZIONE: prendi(x,y),  
PRECONDIZIONI: Posizione(Shakey,y), Posizione(x,y), Pinza(Shakey,vuota)  
EFFETTI: Pinza(Shakey,x),  $\neg$ Posizione(x,y),  $\neg$ Pinza(Shakey,vuota))

Operazione(AZIONE: lascia(x,y),  
PRECONDIZIONI: Pinza(Shakey,x), Posizione(Shakey,y)  
EFFETTI: Pinza(Shakey,vuota), Posizione(x,y),  $\neg$ Pinza(Shakey,vuota))

La soluzione può essere rappresentata dal piano seguente:

vai(Stanza3, Stanza2), prendi(OggettoA, Stanza2), vai(Stanza2, Stanza1), lascia(OggettoA, Stanza1), vai(Stanza1, Stanza4)

Una soluzione deve avere le seguenti caratteristiche: (1) deve essere *efficace* cioè deve garantire il raggiungimento dell'obiettivo; (2) deve essere *completa* nel senso che le precondizioni necessarie per l'esecuzione di ciascuna azione devono essere verificate e, ove necessario, attuate attraverso l'esecuzione delle azioni precedenti; (3) deve essere *consistente* nel senso che non ci debbono essere contraddizioni derivanti dall'ordine di esecuzione delle azioni o dall'istanziamento delle variabili.

Per ottenere una soluzione come quella descritta sopra è necessario definire un algoritmo (algoritmo di pianificazione) che, in base allo stato iniziale, all'obiettivo, e alle azioni disponibili costruisca progressivamente un piano con le caratteristiche di efficacia, completezza e consistenza descritte sopra.

L'algoritmo di pianificazione, per es., può iniziare con un piano formato da una operazione iniziale che ha come effetto lo stato iniziale del mondo e una operazione finale che ha come precondizione l'obiettivo del piano (vedi sotto) e cercare progressivamente di estenderlo inserendo all'interno altre operazioni che rendano il piano via via più completo e consistente fino ad arrivare ad una soluzione.

Operazione(AZIONE: Inizio()),  
EFFETTI: Posizione(Shakey, Stanza3), Posizione(OggettoA, Stanza2))  
Operazione(AZIONE: Fine()),  
PRECONDIZIONI: Posizione(Shakey, Stanza4), Posizione(OggettoA, Stanza1))

In particolare l'algoritmo può procedere cercando di inserire delle azioni che siano in grado di rendere verificate le precondizioni insoddisfatte. Ad es., l'algoritmo può decidere di inserire tra le azioni Inizio() e Fine() l'azione Lascia(OggettoA, Stanza1) allo scopo di rendere vera la precondizione Posizione(OggettoA, Stanza1). Naturalmente l'aggiunta di questa azione nel piano può implicare che nuove condizioni debbano essere verificate (in particolare Shakey deve trovarsi nella stanza n.1 prima che tale azione venga eseguita).

Un piano parziale come questo può essere elaborato seguendo diverse strade alternative alcune delle quali possono rivelarsi dei vicoli ciechi. In questo caso l'algoritmo di pianificazione deve tornare indietro e provare un'altra strada. Si noti come la possibilità di inserire nuove azioni dove è necessario senza alcun vincolo, a differenza del problem solving, può ridurre significativamente lo spazio della ricerca. Nel caso illustrato sopra, ad es., non è necessario prendere in considerazione tutte le possibili azioni che potrebbero essere svolte da Shakey a partire dallo stato iniziale. Shakey può decidere di lasciare l'oggetto A nella stanza n.1 prima di decidere come arrivarci e come prendere l'oggetto.

Algoritmi di planning semplici come questo possono risolvere solo problemi relativamente semplici. Per affrontare problemi più complessi è necessario utilizzare algoritmi di ricerca mirati e linguaggi di rappresentazione più potenti (Russell e Norvig, 1995). Per quanto riguarda il problema del linguaggio di rappresentazione, sono stati elaborati formalismi più complessi che permettono di:

- (1) specificare piani a diversi livelli di descrizione (un livello alto in cui vengono indicate una sequenza di macro-azioni e uno o più livelli più bassi in cui ciascuna macro-azione viene tradotta in una sequenza di azioni più semplici);
- (2) gestire il tempo (ad es. gestire il fatto che ciascuna azione ha una durata o che alcune azioni devono essere svolte entro un certo tempo limite);
- (3) tenere conto di limiti di gestione (ad es. il fatto che ci possono essere dei limiti sul numero e il tipo di attività che si possono svolgere in contemporanea).

La pianificazione è stata utilizzata con successo in diverse aree come ad es. nella pianificazione di missioni spaziali. Tuttavia in altre aree, per es. il controllo di robot mobili come Shakey, ha prodotto risultati deludenti. Ciò è dovuto ad una serie di ragioni:

- Come abbiamo detto all'inizio il pianificatore non si occupa direttamente di come lo stato attuale del mondo gli venga fornito in input e di come il piano prodotto venga effettivamente tradotto in pratica (queste due funzioni devono essere svolte dal sistema percettivo e motorio rispettivamente). Inoltre, abbiamo visto che il planning si aspetta di ricevere in ingresso e produce in uscita delle rappresentazioni di tipo simbolico. Purtroppo estrarre dal mondo attraverso dei sensori fisici una descrizione simbolica dell'ambiente esterno è tutt'altro che ovvio e può rivelarsi una impresa impossibile se ci si aspetta una rappresentazione esatta e completa dell'ambiente esterno. Ciò è dovuto ad una serie di difficoltà: i sensori sono in grado di misurare informazioni quantitative non simboliche, i valori misurati dai sensori sono una indicazione molto indiretta e rumorosa delle quantità misurate, il mondo reale è pieno di ambiguità e spesso in continuo cambiamento, le informazioni estratti dai sensori sono di tipo egocentrico mentre il pianificatore si aspetta una rappresentazione di tipo allocentrico. Dal punto di vista motorio i problemi sono simili. Quando le informazioni fornite in ingresso sono incomplete, rumorose, e parzialmente errate non c'è alcuna garanzia che il piano sviluppato dal pianificatore sulla base di esse sia efficace e, nella misura in cui il sistema percettivo e motorio risultano delle scatole nere inaccessibili al pianificatore, risulta estremamente difficile compensare eventuali malfunzionamenti.
- Il secondo problema riguarda il fatto che se l'ambiente si modifica durante l'esecuzione del piano non c'è alcuna garanzia che il piano possa avere successo visto che il piano viene sviluppato sulla base di una descrizione del mondo valida prima dell'esecuzione del piano stesso. Questo problema è particolarmente presente in ambienti fortemente dinamici come ad es. nel caso di Robocup (Kitano et al., 1998, si veda pure <http://www.robocup.org/>) una competizione tra laboratori di ricerca di tutto il mondo che consiste nello sviluppare squadre di robot calciatori che si fronteggiano in un torneo annuale.
- Infine il terzo problema consiste nel fatto che operare in un ambiente reale spesso richiede di prendere decisioni in tempi brevi (per es. per evitare un ostacolo un robot mobile può dover modificare la propria traiettoria nel giro di poche decine o centinaia di millisecondi. Un tempo che può essere troppo breve per un pianificatore nonostante le attuali potenze di calcolo dei computer).

La ricerca attuale, in particolare nell'ambito della robotica, è proprio incentrata nel tentativo di sviluppare nuove architetture e tecniche di pianificazione (anche basate su formalismi diversi come le reti neurali e i sistemi fuzzy) che possano rispondere in modo più efficace a questi problemi. Alcune linee di ricerca inoltre stanno esplorando la possibilità di sviluppare sistemi ibridi che includono, oltre ad un sistema di pianificazione, un sistema reattivo basato sulle tecniche che descriveremo nella sezioni seguenti.

ESERCIZIO 5.1. Siano dati quattro cubi,  $C_1, C_2, C_3, C_4$  disposti sul tavolo  $T$  nel modo seguente. I cubi  $C_3$  e  $C_4$  sono poggiati su  $T$ ,  $C_1$  è poggiato su  $C_4$ , e  $C_2$  è poggiato su  $C_3$ . I cubi  $C_1$  e  $C_2$  sono liberi, vale a dire non ci sono cubi posti sopra di essi. Si assuma cioè che lo stato iniziale del mondo sia il seguente:

sopra( $C_3, T$ ), sopra( $C_4, T$ ), sopra( $C_1, C_4$ ), sopra( $C_2, C_3$ ), libero( $C_1$ ), libero( $C_2$ )

Le operazioni disponibili sono:

Operazione(*azione*: sposta( $x, y$ ),  
*precondizioni*: libero( $x$ ), libero( $y$ ),  
*effetti*: sopra( $x, y$ ));

Operazione(*azione*: sgombra( $x$ ),  
*precondizioni*: sopra( $x, z$ ),

*effetti: sopra(x,T),libero(z);*

Si scriva un piano che trasferisca il cubo  $C_1$  su  $C_3$  cioè un piano che permetta di raggiungere lo stato seguente:

sopra( $C_1, C_3$ )

---

## 2. La robotica Behavior-Based e gli approcci "ibridi"

Le difficoltà descritte sopra hanno spinto Brooks (1991) a proporre una visione radicalmente diversa da quella dell'intelligenza artificiale classica descritta sopra. Secondo Brooks un sistema intelligente deve essere visto come un sistema dotato di un corpo (che include un sistema sensoriale e un sistema motorio), deve essere situato in un ambiente esterno e deve poter interagire direttamente con tale ambiente (piuttosto che con delle rappresentazioni simboliche elaborate da uno sperimentatore umano). L'architettura fondamentale di un sistema di questo tipo non deve essere organizzata in una sequenza di moduli funzionali che vengono attivati in successione (Figura 1, sinistra) come nella visione classica ma piuttosto in una serie di moduli comportamentali collegati in modo diretto all'ambiente esterno (Figura 1, destra).

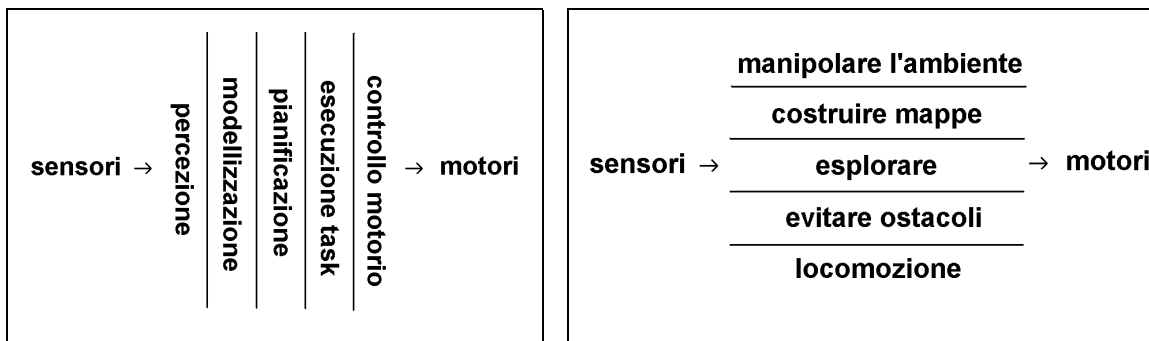


Figura 1. **Sinistra:** La decomposizione operata nell'approccio classico. **Destra:** La decomposizione proposta da Brooks. (Figura riprodotta da Brooks, 1991).

Così come nella visione classica dunque, il problema di costruire un agente intelligente viene affrontato utilizzando una tecnica di decomposizione (il problema complessivo che il sistema deve affrontare e conseguentemente l'architettura interna del sistema viene suddivisa in una serie di moduli in grado di risolvere i diversi sotto-problemi identificati). La differenza consiste nel modo in cui viene effettuata la decomposizione. Nel caso dell'approccio behavior-based ci si basa sul comportamento. Il comportamento che il sistema deve essere in grado di produrre viene suddiviso in una serie di sotto-comportamenti o comportamenti semplici (eventualmente procedendo con una suddivisione di tipo gerarchico). Una volta che la decomposizione è stata effettuata, si procede sviluppando una serie di strati o moduli in grado di produrre i sotto-comportamenti identificati che vengono aggiunti in modo incrementale.

Il sistema finale dunque consiste in una serie di moduli-comportamentali (come ad es. "Evita gli ostacoli" che serve ad evitare eventuali collisioni oppure "Torna al nido" che consente al sistema di raggiungere una certa zona dell'ambiente). I moduli possono essere implementati con delle procedure o con altri sistemi (ad es. reti neurali) e tipicamente ricevono in ingresso lo stato corrente di alcuni sensori (ad es. telecamera, sensori ad

infrarosso, sensori di contatto), alterano lo stato corrente di alcuni attuatori del sistema (ad es. i motori che controllano le ruote o le zampe del sistema oppure un eventuale braccio meccanico) ed eventualmente lo stato di alcune variabili o stati interni del sistema. Tali moduli interagiscono anche tra di loro (ad es. un modulo comportamentale può attivare o inibire un altro modulo comportamentale). Complessivamente dunque, un sistema di questo tipo consiste in una rete di moduli comportamentali che interagiscono tra di loro.

Questo diverso modo di operare ha una serie di conseguenze che rendono i sistemi behavior-based molto diversi dai sistemi dell'intelligenza artificiale classica descritti nella sezione precedente.

Una prima differenza consiste nel fatto che questa tecnica permette di procedere in modo incrementale sviluppando prima i moduli più basilari e aggiungendo poi progressivamente altri moduli. Ciò consente di sviluppare uno strato per volta verificandone l'efficacia ed eventualmente introducendo delle modifiche prima di procedere allo sviluppo di altri moduli. Ciascun modulo infatti è un sistema in grado di produrre autonomamente un certo comportamento anche se può basarsi sulla presenza di moduli più basilari sviluppati precedentemente (per es., un modulo incaricato di produrre un comportamento di esplorazione non deve occuparsi direttamente di evitare gli ostacoli se esiste un modulo, sviluppato precedentemente, in grado di farlo).

Una seconda differenza consiste nel fatto che non è prevista una rappresentazione interna esplicita e completa dell'ambiente esterno. Ciascun modulo, ove necessario, può basarsi su delle proprie rappresentazioni interne o meglio su dei propri stati interni. Tuttavia, poiché i moduli sono orientati a produrre dei comportamenti specifici, ciascun modulo tende ad avere rappresentazioni parziali (che includono solo le caratteristiche dell'ambiente esterno necessarie per produrre il comportamento in questione) e finalizzate all'azione (cioè rappresentazioni finalizzate alla produzione di un comportamento specifico).

Una terza differenza consiste nel fatto che sia gli strati più bassi che quelli più alti interagiscono direttamente con l'ambiente esterno e che la quantità di elaborazioni interne necessarie per la produzione del comportamento desiderato sono di norma assai contenute.

Infine, una quarta differenza consiste nel fatto che in questo approccio viene attribuito un ruolo fondamentale all'emergenza. Il comportamento complessivo del sistema non è visto come qualcosa che viene prodotto dal sistema direttamente ma piuttosto come qualcosa che emerge dall'interazione tra le sotto-componenti del sistema e tra il sistema e l'ambiente. Ad es. un comportamento che consente al robot di inseguire una fonte luminosa in movimento può emergere dall'interazione tra due moduli comportamentali del sistema che consentono al robot di evitare gli ostacoli e di avvicinarsi ad una fonte luminosa.

L'approccio behavior-based è stato applicato con successo nello sviluppo sia di agenti software che di robot mobili in grado di svolgere compiti relativamente semplici in modo autonomo (come ad es. navigare verso un obiettivo evitando gli ostacoli). Al momento tuttavia non è chiaro se questo approccio possa essere utilizzato con successo per affrontare compiti significativamente più complessi. La risposta potrebbe arrivare dallo stesso Brooks che recentemente ha lanciato un progetto denominato COG (Brooks, 1997) il cui obiettivo è quello di utilizzare questo approccio per sviluppare un robot umanoide in grado di esibire forme di intelligenza paragonabili a quelle di un bambino (i primi risultati sono incoraggianti anche se il progetto sembra procedere molto più lentamente del previsto). Dal punto di vista applicativo questi sistemi sono stati utilizzati per lo sviluppo di veicoli spaziali e di robot in grado di camminare, manipolare oggetti e esplorare ambienti ostili.

L'approccio behavior-based è stato spesso accusato di ignorare l'importanza delle rappresentazioni interne e di basarsi quasi esclusivamente su meccanismi reattivi (cioè su sistemi in cui, come nei veicoli di Braitenberg, lo stato dei sensori influenza in modo diretto lo stato dei motori). In realtà questa critica può essere rovesciata e si può sostenere che tale

approccio ha giustamente riportato l'attenzione sull'importanza dei meccanismi reattivi e sul fatto che tali meccanismi possono svolgere un ruolo fondamentale anche in sistemi in grado di esibire forme di intelligenza elevate. Il limite di tale approccio potrebbe consistere piuttosto in una difficoltà intrinseca del ricercatore di progettare sistemi che sfruttino l'emergenza (per un tentativo di identificare esplicitamente dei criteri che permettano di creare un'ingegneria dell'emergenza si veda Pfeifer & Scheier, 1999). Un modo per affrontare questo problema, come vedremo nella sezione successiva, è quello di delegare la progettazione del sistema ad un processo automatico basato su un algoritmo evolutivo.

---

ESERCIZIO 5.2. Perché l'approccio behavior-based, al contrario dell'approccio "classico" basato sulla pianificazione, permette di sviluppare un sistema in modo incrementale ?

---

Parallelamente all'approccio behavior-based si è affermata una nuova area di ricerca che ha proposto di utilizzare sistemi ibridi composti da un pianificatore centrale (del tipo di quelli che abbiamo descritto nella sezione precedente) e da un sistema reattivo (cioè un sistema che opera a livello sub-simbolico e in cui i sensori sono direttamente connessi con i motori). Questi sistemi sono tipicamente organizzati in tre parti funzionalmente distinte: (1) il pianificatore; (2) il sistema reattivo; (3) il sistema di interfaccia che permette ai due sistemi di comunicare superando i problemi di scala temporale e di relazione tra i diversi tipi di rappresentazione utilizzati (si veda ad es. Arkin, 1989).

Con questo tipo di architetture si tenta di affrontare i problemi lasciati irrisolti dai sistemi di pianificazione che abbiamo ricordato nel paragrafo precedente, quei problemi che si pongono quando l'azione del sistema avviene in base ad informazioni incomplete o in presenza di rumore o deve svolgersi in tempo reale. Ciò è particolarmente evidente quando un robot deve agire in un ambiente fortemente dinamico. Un laboratorio di esperienze in questo ambito è dato da *RoboCup*, una competizione tra i laboratori di ricerca di tutto il mondo che mettono in campo squadre di robot calciatori che si fronteggiano in tornei annuali (Kitano et al., 1998). Questo tipo di ricerche permette di sperimentare anche il coordinamento dell'azione di più sistemi che cooperano tra di loro, un problema affrontato in generale nell'ambito dei sistemi multiagente (sui quali cfr. CAP. 9).

Le architetture ibride sfruttano sistemi di controllo e di rappresentazione della conoscenza molto diversi. Si va da sistemi che usano logiche basate sul ragionamento di senso comune (cfr. CAP.3) a sistemi che usano logiche fuzzy sia nel controllo a livello reattivo sia nella pianificazione. L'approccio ibrido ha abbandonato l'idea che la pianificazione e il ragionamento debbano svolgersi sulla base di un modello completo del mondo separato dalla percezione e dall'azione, ma ha conservato l'idea che un modello del mondo, comunque parziale o da aggiornare continuamente nell'interazione con l'ambiente, sia necessario quando un robot debba svolgere compiti di previsione e di elaborazione di azioni di una certa complessità (cfr. Bonasso, Dean, 1997, per una rassegna di architetture variamente ibridie e di diversi sistemi di controllo).

L'approccio ibrido si propone di combinare i vantaggi dei sistemi reattivi (in termini di robustezza, capacità di rispondere in tempo reale e di adattarsi ai cambiamenti ambientali) con i vantaggi dei sistemi di pianificazione (generalità, ragionamento). I limiti dei sistemi ibridi di questo tipo sono dovuti principalmente alla difficoltà di far dialogare il sistema reattivo e il pianificatore e in particolare al problema del *symbol grounding* (Harnad, 1990) cioè al problema di trovare la relazione tra i simboli utilizzati dal pianificatore e le informazioni sub-simboliche registrate dai sensori.

### 3. La Robotica Evolutiva

L'approccio behavior-based si basa sull'assunto implicito che la decomposizione del comportamento desiderato in una serie di comportamenti basilari corrispondenti a diversi strati del sistema di controllo produca una semplificazione del problema complessivo. Tale assunzione tuttavia appare problematica da un punto di vista teorico.

Per illustrare questo punto è necessario distinguere due modi di descrivere il comportamento di un sistema situato in un ambiente:

- Una descrizione *distale* è una descrizione dal punto di vista di un osservatore esterno che utilizza una etichetta linguistica (ad es. esplorare l'ambiente) per descrivere una sequenza di cicli senso-motori prodotti da un sistema (naturale o artificiale) o che il sistema deve produrre.
- Una descrizione *prossimale* è una descrizione dal punto di vista del sistema che indica come tale sistema reagisce al flusso di informazioni sensoriali che gli arrivano dall'ambiente esterno

Il comportamento da un punto di vista distale non è semplicemente il risultato del comportamento da un punto di vista prossimale ma è il risultato dell'interazione tra il comportamento prossimale e l'ambiente. Ciò implica che non c'è una relazione diretta tra il comportamento di un agente a livello prossimale e distale. Per es. comportamenti che appaiono complessi da un punto di vista distale possono in realtà essere prodotti da un sistema di controllo (che opera a livello prossimale) estremamente semplice.

Il problema dell'approccio behavior-based è che la decomposizione che viene utilizzata per determinare l'architettura generale del sistema di controllo che opera a livello prossimale viene effettuata dallo sperimentatore sulla base di una descrizione distale del comportamento desiderato. Poiché, come abbiamo detto, non c'è una relazione diretta tra livello prossimale e livello distale non c'è alcuna garanzia che le decomposizioni che appaiono sensate a livello distale siano efficaci al livello prossimale. Di fatto si possono identificare dei casi in cui decomposizioni che appaiono ragionevoli da un punto di vista distale non solo non garantiscono una semplificazione del problema complessivo ma possono implicare la creazione di sotto-obiettivi più complessi dell'intero problema originario (Nolfi e Floreano, 2000).

Un modo per superare questo tipo di problema è quello di utilizzare sistemi in grado di auto-organizzarsi (eventualmente ricorrendo ad una suddivisione del problema in sotto-problemi) senza richiedere una fase in cui lo sperimentatore effettui la decomposizione del sistema. Ciò può essere ottenuto, ad es., utilizzando le metodologie sviluppate nell'ambito della cosiddetta robotica evolutiva (Nolfi e Floreano, 2000) che rappresenta il tentativo di sviluppare dei robot attraverso un processo analogo all'evoluzione naturale. L'idea di base di questo metodo è molto semplice: (1) viene creata una popolazione di agenti dotati di sistemi di controllo (ed eventualmente corpi) casualmente diversi; (2) vengono fatti "riprodurre" gli agenti il cui comportamento approssima maggiormente quello desiderato (viene cioè creata una nuova popolazione simile ma non identica ai migliori individui della generazione precedente); (3) si ripetono le fasi 1 e 2 per certo numero di generazioni finché non si ottiene almeno un individuo in grado di produrre il comportamento desiderato. Si noti che in questo approccio allo sperimentatore non è richiesto di indicare come decomporre il sistema di controllo del robot che si intende sviluppare o quali tipi di comportamenti basilari devono essere prodotti dal sistema per risolvere il problema complessivo ma semplicemente un criterio per valutare quanto un certo comportamento approssima nel complesso il comportamento desiderato.



Il robot dunque viene visto come un sistema che sviluppa le proprie abilità autonomamente interagendo con l'ambiente esterno e non come un sistema che viene disegnato dallo sperimentatore. I sistemi sviluppati attraverso questo metodo di norma: (1) sono in grado di risolvere problemi relativamente complessi da un punto di vista distale utilizzando delle strategie estremamente semplici dal punto di vista prossimale e (2) spesso identificano delle soluzioni qualitativamente diverse da quelle che possono essere elaborate da un ricercatore o un ingegnere che cerca di progettare esplicitamente un robot in grado di risolvere il problema dato.

Per illustrare questo punto consideriamo ad es. il caso di un semplice robot dotato di sensori ad infrarosso in grado di segnalare la presenza di ostacoli ad una distanza massima di circa 4 cm e di due ruote in grado di ruotare in entrambe le direzioni a velocità diverse. Immaginiamo che tale robot sia posto in una serie di ambienti rettangolari delimitati da mura in cui le pareti nord e sud siano sempre più lunghe delle pareti est e ovest ma in cui la dimensione assoluta delle mura e la proporzione tra le mura più lunghe e quelle più corte vari di volta in volta. Infine immaginiamo che il robot venga posto inizialmente al centro di tale ambiente rettangolare con un orientamento casuale e che gli venga chiesto di raggiungere e rimanere nella prossimità dell'angolo nord-ovest o dell'angolo sud-est ma non nei due altri angoli.

Se seguiamo l'approccio Behavior-Based la prima cosa che dobbiamo fare è suddividere il comportamento desiderato (raggiungere e rimanere nell'angolo nord-ovest o sud-est) in una serie di comportamenti basilari semplici.

Un modo possibile di procedere ad es. potrebbe essere quello di postulare che per risolvere tale compito il robot debba essere in grado di: (1) esplorare l'ambiente costruendo una rappresentazione interna della forma dell'ambiente (mappa); (2) aggiornare continuamente una rappresentazione interna che indichi quale sia la posizione relativa del robot rispetto alla mappa; (3) una volta completata l'estrazione della mappa, navigare verso uno dei due angoli obiettivo basandosi sulla mappa dell'ambiente e sulla rappresentazione della posizione relativa del robot nell'ambiente. Questo primo modo di risolvere il problema richiede lo sviluppo di un sistema controllo organizzato in tre strati (moduli) relativamente complessi. Inoltre la possibilità di sviluppare dei moduli in grado di risolvere i problemi 1 e 2 appare tutt'altro che scontata se teniamo conto del fatto che l'avanzamento effettivo del robot dato un certo comando non è mai esattamente quello previsto (a causa di una serie di fattori imprevedibili come ad es. la presenza di piccole asperità del terreno). Tali piccole discrepanze tra la variazione di posizione stimata dal robot e la variazione effettiva accumulandosi nel tempo tendono a portare ad una discrepanza significativa tra la posizione stimata dal robot e la sua posizione effettiva e ciò può impedire al robot sia di estrarre dall'ambiente una mappa effettivamente rispondente alla realtà esterna sia di utilizzare tale mappa per navigare verso la direzione corretta.

Un altro possibile modo di procedere consiste nell'immaginare una soluzione del problema che si basi su una rappresentazione interna limitata a quegli aspetti che sono necessari per la risoluzione del compito piuttosto che su una rappresentazione completa ed esplicita dell'ambiente esterno. Ad es. possiamo immaginare che il robot possa risolvere il compito: (1) esplorando l'ambiente fino ad incontrare una parete; (2) circumnavigando l'ambiente mantenendo le pareti sul proprio lato destro; (3) misurando la lunghezza del muro corrente e confrontandola con quella del muro incontrato precedentemente; (4) fermandosi di fronte ad un angolo quando la lunghezza del muro corrente è maggiore di quella del muro incontrato precedentemente. Questo tipo di strategia può essere implementata in modo molto più semplice della precedente visto che richiede l'estrazione dall'ambiente di informazioni egocentriche (la lunghezza del muro corrente) e non allocentriche come nel caso della mappa dell'ambiente descritta sopra. Si noti inoltre come questo tipo di strategia utilizzi direttamente

il mondo esterno e non un suo modello ricostruito internamente per risolvere il problema. Invece di effettuare delle operazioni di misura sulla mappa dell'ambiente precedentemente estratta infatti, il robot "misura" la lunghezza delle pareti muovendosi nell'ambiente (cioè esibendo un comportamento di circumnavigazione).

E' interessante che, utilizzando l'approccio evolutivo descritto sopra (cioè evolvendo una popolazione di sistemi di controllo e selezionando gli individui migliori sulla base nel numero di volte in cui tali individui sono in grado di raggiungere uno dei due angoli obiettivo) possiamo osservare l'emergere di un terzo modo di risolvere questo compito ancora più semplice delle due strategie descritte sopra (Nolfi, in stampa). Dopo un certo numero di generazioni emergono degli individui che: (1) esplorano l'ambiente finché raggiungono un angolo dell'ambiente; (2) abbandonano l'angolo con una direzione di circa 45 gradi rispetto alle due pareti che formano tale angolo; (3) quando incontrano una nuova parete con un angolo di circa 45 gradi sulla propria destra seguono la parete fino a raggiungere l'angolo corrispondente mentre quando incontrano una parete con un angolo di circa 45 gradi sulla propria sinistra seguono semplicemente la parete fino a raggiungere l'angolo corrispondente (Figura 2).

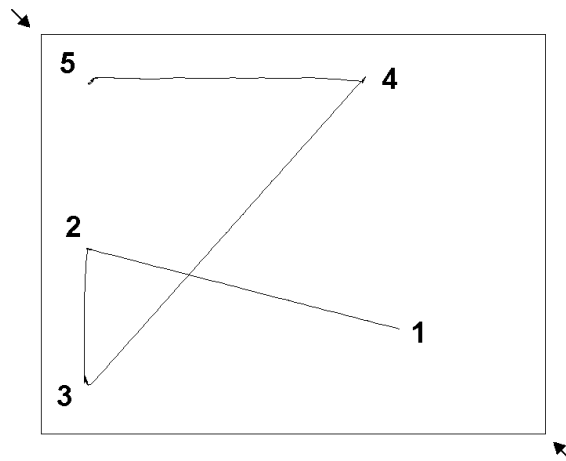


Figura 2. Il comportamento di uno dei robot evoluti. Il rettangolo indica l'arena del robot circondata da mura. Le frecce indicano le due aree obiettivo. Le linee all'interno del rettangolo indicano la traiettoria del robot. In questo esempio (1) il robot parte dalla posizione indicata con il numero 1, (2) incontra un primo muro e lo evita, (3) incontra l'angolo sud-ovest dell'ambiente e lo abbandona con una traiettoria di circa 45 gradi rispetto alle due mura che formano l'angolo, (4) incontra una parete con un angolo di circa 45 gradi sul proprio lato sinistro e gira in senso anti-orario, (5) raggiunge l'angolo nord-ovest (cioè una delle due aree obiettivo).

Questo tipo di strategia è ancora più semplice della precedente in quanto non richiede di "misurare" la lunghezza di due mura adiacenti e di confrontare gli stati interni corrispondenti alle due lunghezze. In un certo senso si può dire che questi robot, attraverso il loro comportamento motorio, sono in grado di misurare direttamente la *lunghezza relativa* di due mura adiacenti. Infatti, dato il comportamento esibito dal robot (abbandonare un angolo dell'ambiente con un angolo di 45 gradi), la presenza di uno stimolo sensoriale corrispondente alla situazione "muro a 45 gradi sulla sinistra" è un indice inequivocabile che il muro in prossimità del robot è uno dei due muri lunghi dell'ambiente (e di un muro corto nel caso dello stimolo "muro a 45 gradi sulla destra"). Si noti come tale stimolo di per se (cioè prescindendo dal comportamento del robot) non indichi alcuna informazione sulla lunghezza del muro locale ne tantomeno sulla sua lunghezza relativa delle mura adiacenti. E' il comportamento esibito dal robot che in un certo senso conferisce a questo stimolo un significato così informativo. In altre parole non solo questi robot tendono ad utilizzare

direttamente l'ambiente (invece che un modello di esso) per risolvere un compito, ma addirittura possono conferire attraverso il comportamento un potere rappresentativo agli stati sensoriali esterni.

La strategia descritta nella Figura 2 richiede uno stato interno. Per reagire correttamente agli stimoli corrispondenti alla situazione "parete a 45 gradi sulla destra o sulla sinistra" il robot deve "sapere" se ha precedentemente abbandonato un angolo dell'ambiente. Il tipo di rappresentazione richiesta tuttavia è fortemente incompleta e indiretta (codifica ciò che il robot ha fatto e non direttamente quali sono le proprietà dell'ambiente).

Questo tipo di strategia è in grado di raggiungere l'obiettivo in qualsiasi tipo di ambiente rettangolare indipendentemente dalle dimensioni assolute e dalle dimensioni relative tra mura lunghe e corte ed è quindi una strategia generale altrettanto efficace delle strategie descritte sopra. In conclusione, è interessante osservare che:

- Il sistema sfrutta l'interazione con l'ambiente esterno per risolvere il problema. Di conseguenza il comportamento osservato non è riconducibile in modo diretto al sistema di controllo. Il comportamento emerge dall'interazione tra il robot e l'ambiente.
- Il tipo di soluzioni sviluppate dagli individui sottoposti al processo di evoluzione artificiale sono qualitativamente diverse dal tipo di soluzioni che possono essere ideate da un ingegnere o da un ricercatore che si pone l'obiettivo di costruire un sistema in grado di risolvere questi problemi. In altre parole questo tipo di soluzioni estremamente semplici in cui il comportamento emerge dall'interazione tra il sistema e l'ambiente esterno sono difficili da disegnare dal punto di vista di un osservatore esterno.

L'approccio evolutivo è stato applicato con successo nello sviluppo sia di agenti software che di robot mobili in grado di svolgere compiti relativamente semplici in modo autonomo (come ad es. esplorare un ambiente, camminare, navigare verso un obiettivo evitando gli ostacoli, discriminare oggetti). Si tratta di un area di ricerca piuttosto recente e di conseguenza è possibile che i limiti di questo approccio non siano stati ancora chiaramente identificati. Da un punto di vista applicativo in particolare non è chiaro quale sia il limite di complessità che possa essere raggiunto.

---

ESERCIZIO 5.3. Immaginate di voler costruire il sistema di controllo di un organismo artificiale che vive in un ambiente circolare come quello descritto nella Figura 3 e immaginate che il lato ovest dell'ambiente sia significativamente più ricco di cibo del lato est (immaginate cioè che l'obiettivo dell'organismo sia raggiungere e rimanere nel lato ovest).

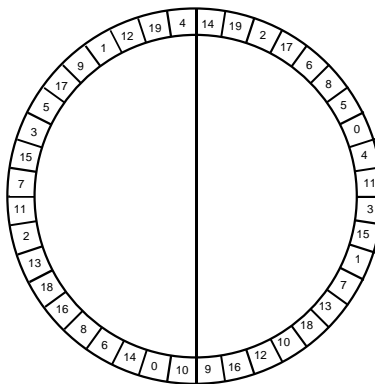


Figura 3. L'ambiente dell'organismo.

Il corpo dell'organismo ha una dimensione paragonabile a quella di una cella, è dotato di due zampe (che gli permettono di spostarsi di una cella in senso orario o anti-orario oppure di stare fermo) e di un sensore posto sulla parte inferiore del corpo che gli permette di identificare il colore della cella. Le

celle hanno colori diversi (che nella figura sono indicati con numeri diversi da 0 a 19) ma, come spesso accade nel mondo reale, l'informazione veicolata dai sensori è ambigua in quanto un certo colore può trovarsi in entrambe le aree dell'ambiente (in questo caso l'ambiguità è totale in quanto ciascun colore è rappresentato una volta sia nella parte est che nella parte ovest).

Ora cercate di identificare quale è il sistema di controllo più semplice possibile che permetta all'organismo, che può trovarsi inizialmente in una cella qualsiasi dell'ambiente, di raggiungere e rimanere nella zona ovest.

---

## Conclusioni

In questo capitolo abbiamo introdotto il lettore a quattro diverse architetture di sistema intelligente, basate rispettivamente sull'approccio cosiddetto "classico" basato sulla pianificazione, sull'approccio behavior-based, sull'approccio ibrido e l'approccio evolutivo. Abbiamo discusso i problemi legati alla assunzione che il sistema debba avere un modello completo ed esplicito del mondo nel quale deve operare e la possibilità di utilizzare invece rappresentazioni parziali ed implicite dell'ambiente esterno. Infine, abbiamo discusso del ruolo dell'ambiente e di come il sistema possa esibire comportamenti intelligenti che risultano dall'interazione tra le componenti interne del sistema e tra il sistema e l'ambiente esterno.

## Ringraziamenti

L'autore ringrazia Angelo Oddi per i suggerimenti e commenti forniti.

## Bibliografia

- Bonasso P., Dean T. (1997). A retrospective of the AAAI Robot Competitions. *AI Magazine*, 18(1):11-23.
- Braitenberg V. (1984). I veicoli pensanti. Garzanti. Milano.
- Brooks R.A. (1991). New approaches to Robotics. *Science*, 253:1227-1232.
- Brooks, R.A. (1997). The Cog Project, *Journal of the Robotics Society of Japan*, (15) 7:968-970.
- Fikes, R. E. & Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 5(2): 189-208.
- Harnad S. (1990) Symbol grounding problem. *Physica D*, 42:335-464.
- Kitano H., Asada M., Kuniyoshi Y., Noda I., Osawa E., Matsubara H. (1998). RoboCup: A Challenge Problem of AI. *AI Magazine*, 18:73-85.
- Nolfi S., and Floreano D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Cambridge, MA: MIT Press/Bradford Books
- Pfeifer R. & Scheier C. (1999) *Understanding Intelligence*. Cambridge, MA:MIT Press
- Russell S.J & Norvig P. (1995). *Artificial Intelligence: A modern Approach*. Englewood Cliffs (NJ): Prentice Hall.

## Soluzioni Esercizi

- 5.1 Il piano che soddisfa la richiesta è: sgombra( $C_1$ ), sgombra( $C_2$ ),sposta( $C_3,C_1$ ).
- 5.2 I sistemi behavior-based sono costituiti da una gerarchia di moduli comportamentali che possono funzionare autonomamente. Qualora si voglia sviluppare un robot in grado di evitare gli ostacoli ed esplorare un ambiente è possibile per es. prima implementare il modulo che consente al robot

di evitare gli ostacoli e poi, dopo aver verificato che il robot si comporta in modo corretto, passare alla implementazione del modulo di esplorazione. L'implementazione di questo secondo modulo non potrà alterare la capacità del robot di evitare gli ostacoli. Nel caso dell'approccio basato sulla pianificazione invece qualsiasi modifica interna del sistema (ad es. nel repertorio delle azioni disponibili) può alterare il comportamento del sistema in modo generalizzato.

- 5.3 Utilizzando il metodo evolutivo descritto nella sezione 3 si può osservare che esistono un gran numero di soluzioni puramente reattive (che non richiedono cioè alcuna elaborazione interna). Una di queste soluzioni consiste nel muoversi in senso orario nel caso delle celle 1 3 7 9 10 11 12 13 15 16 18 e in senso antiorario nei casi rimanenti. Disegnate delle frecce orientate nel senso orario o anti-orario indicato a fianco delle 40 celle dell'ambiente rappresentato nella Figura 2 e immaginate che l'organismo si muova seguendo l'orientamento delle frecce. Come vedrete questa strategia permette all'organismo di abbandonare la parte est e rimanere nella parte ovest oscillando tra due celle adiacenti in uno dei quattro attrattori che si formano nella zona sinistra dell'ambiente (nelle celle [10-0], [13-2], [3-5], [19-4]). Un attrattore nel comportamento dell'organismo in questo caso consiste di due celle adiacenti a cui l'organismo risponde in senso orario e anti-orario.

Notate come questa soluzione non richieda alcuna rappresentazione interna dell'ambiente né alcuna memoria. Notate inoltre come la capacità di discriminare la zona giusta dell'ambiente non è il risultato di una "decisione" del sistema conseguente ad un certo stimolo sensoriale o ad una certa sequenza di stimoli sensoriali ma è il risultato emergente di una serie di azioni. Per es., il fatto che il sistema si muova in senso orario nel caso delle celle n.7 non ha alcuna funzione di per sé ma assume un significato se si tiene conto di come il sistema risponde in tutti gli altri casi.

Se avete immaginato una soluzione simile a questa (ce ne sono molte) avete un fiuto particolare per scoprire soluzioni "emergentiste". Se invece, come è più probabile, avete immaginato una soluzione più complessa in cui ad es. il sistema si mantiene in memoria l'ultima cella visitata e decide di fermarsi quando la cella corrente e la cella precedentemente visitata corrispondono a due colori definiti non vuol dire che non siete sufficientemente bravi ma semplicemente che, per un osservatore esterno, è estremamente difficile identificare soluzioni che sfruttano l'interazione senso-motoria del sistema con l'ambiente esterno e i comportamenti emergenti come quello descritto in questo esempio.

Questo naturalmente è un caso estremamente semplice e la strategia descritta è fortemente dipendente dalle caratteristiche dell'ambiente all'interno del quale l'organismo è stato evoluto. Per esempi di strategie qualitativamente simili che permettono a dei robot di risolvere compiti complessi anche in condizioni ambientali variabili si veda Nolfi e Floreano (2000).