

La struttura di un sistema esperto e le sue differenze con un programma convenzionale

Credo che il primo sistema esperto sia comparso nelle favole, dove il solito genietto sta nascosto negli stivali delle sette leghe o nella manica dell'alunno che, a scuola, ha bisogno di un piccolo aiuto per svolgere il proprio compito in classe. Mentre scrivo ho vicino a me un mio piccolo amico, Antonio, mio non sempre assiduo allievo in matematica, che di un buon sistema esperto avrebbe proprio bisogno, visto che l'algebra non è proprio il suo forte; vedrò di accontentarlo, un giorno, magari mutuando qualche algoritmo giocoso dalle rubriche di intelliGIOCHI

Geni o no, ognuno di noi adopera sistemi esperti in embrione ogni giorno; a rigore sono sistemi esperti (ma non troppo) le macchine più o meno intelligenti che usiamo ogni giorno; l'ascensore che si rifiuta di tirar su quattro De Masi accatastati nella cabina è, a suo modo, esperto.

È chiaro quindi, anche in base a quanto abbiamo visto la volta scorsa, che un S.E. è strutturato essenzialmente in modo da eseguire continuamente delle scelte, così da giungere, grazie a decisioni euristiche, a conclusioni «il più possibile prossime alla verità». Tutto ciò è tanto più vero quanto più la scelta è meccanizzata; la fotocopiatrice che regola da sé la quantità di toner da sbattere sulla carta o la macchina fotografica che autoregola l'esposizione sono sicuramente meno esperti dell'analizzatore di qualità delle acque della Ferrarelle o della San Pellegrino. Ma in che cosa, al momento della costruzione, un sistema esperto differisce da un programma convenzionale?

Un sistema esperto è differente da un programma convenzionale non solo nella sua modalità di stesura, ma anche nella sua tipologia operativa. La differenza maggiore e più evidente è comunque quella che essi manipolano conoscenza, mentre i programmi costruiti con linguaggi più convenzionali manipolano dati. La Teknowledge, una compagnia che si dedica esclusivamente alla costruzione di sistemi esperti, ha redatto una comparazione accurata e fedele delle differenze di cui abbiamo parlato prima, comparazione riassunta nella figura a.

Un sistema esperto deve avere una

serie di caratteristiche che, riassunte nella figura b, possono essere così elencate:

- esperienza
- ragionamento simbolico
- potenza
- autoconoscenza.

Un S.E. è definito (Brackman, Amarel, Engelman, Engelmores, Feigenbaum, e Wilkins, *What are Expert Systems?*, in «Building Expert System», Addison-Wesley, 1983) quando si applicano compiutamente tutte le qualità dappima elencate; vediamo le caratteristiche di ognuna in maggiore dettaglio.

Esperienza

Un sistema esperto deve funzionare alla perfezione, vale a dire che deve possedere la stessa efficienza di un esperto umano nel campo di interesse comune. Ma produrre soluzioni esatte (nei limiti euristici) non è tutto. Gli esperti umani con sufficiente esperienza non solamente producono buoni risultati, ma lo fanno in maniera rapida, mentre persone preparate, ma non notevolmente impraticate nel mestiere tenderanno ad essere indecise o, comunque, a impiegare più tempo di quello occorrente normalmente. Un sistema esperto deve essere quindi abile, deve cioè applicare la sua conoscenza per produrre soluzioni in maniera affidabile e efficiente, usando efficacemente le scorciatoie che un umano applicherebbe per ridurre o eliminare analisi e calcoli inutili, ridondanti o non strettamente necessari. Per essere pari (si fa per dire) a un esperto umano un sistema esperto deve essere «risoluto», vale a dire che deve raggiungere la sicurezza della risposta, non solo, ma nella maniera più rapida e semplice possibile. Questa operazione di cernita e decisione deve ovviamente essere eseguita a monte del problema, quando l'albero delle scelte non è eccessivamente modificato. Questo ha portato a produrre tecniche diverse per affrontare il problema, tecniche che solo negli ultimi tempi hanno raggiunto una certa omogeneizzazione.

Analisi convenzionale dei dati	Ingegneria della conoscenza
<ul style="list-style-type: none"> • rappresentazione e uso di dati • processi algoritmici di soluzione <ul style="list-style-type: none"> • processi ripetitivi • manipolazione di grandi basi di dati 	<ul style="list-style-type: none"> • rappresentazione e uso di conoscenza <ul style="list-style-type: none"> • processi euristici di soluzione • processi inferenziali • manipolazione di grandi basi di conoscenza

Figura a - Comparazione tra l'attività di un programma convenzionale e di un sistema esperto.

Ragionamento simbolico

Si tratta di una condizione piuttosto semplice da intendere. Quando esperti umani risolvono un problema di quelli tipicamente adottabili da un sistema esperto, lo fanno senza affrontare selve di equazioni matematiche o di calcoli laboriosi. Al contrario essi scelgono dei simboli per rappresentare i concetti del problema e applicano strategie diverse e tecniche euristiche per manipolare tali concetti secondo le loro esigenze e necessità. Un sistema esperto ragiona e lavora esattamente allo stesso modo, rappresentando simbolicamente la conoscenza e lavorando su tali simboli utilizzandoli come gli stessi concetti del problema. In gergo di Alper «simbolo» si intende (Stefik, Bobrow, Mittal, Conway, «*Knowledge Programming in LO-OPS; Report on an Experimental Course*» in A.I. Magazine - 1° quadr. '83) una stringa di caratteri che rappresenta un concetto proprio del mondo esterno: un esempio di simboli è rappresentato di seguito:

computer
costruttore
guasti
3
4
0.75

Questi simboli possono essere combinati tra loro per esprimere relazioni che li legano. Quando queste relazioni sono rappresentate in processi o in un problema di Intelligenza Artificiale, esse sono definite strutture simboliche. Un esempio, che coinvolge i simboli precedentemente definiti è il seguente:

(difettoso computer)
(venduto da produttore)
(1 su 4 guasti)
(affidabilità 0.75)

In termini reali la struttura può essere così tradotta: «Il computer è difettoso» — «il computer è venduto dal produttore» — «su 4 computer se ne guasta 1» — «l'affidabilità dei computer di quel produttore è del 75%».

Ovviamente, per risolvere problemi di tal fatta e giungere a tale conclusione un sistema esperto manipola i simboli precedentemente espressi e non esegue calcoli matematici particolari. La conclusione delle operazioni si configura in una rappresentazione della conoscenza (definibile come scelta, rappresentazione e interpretazione dei simboli

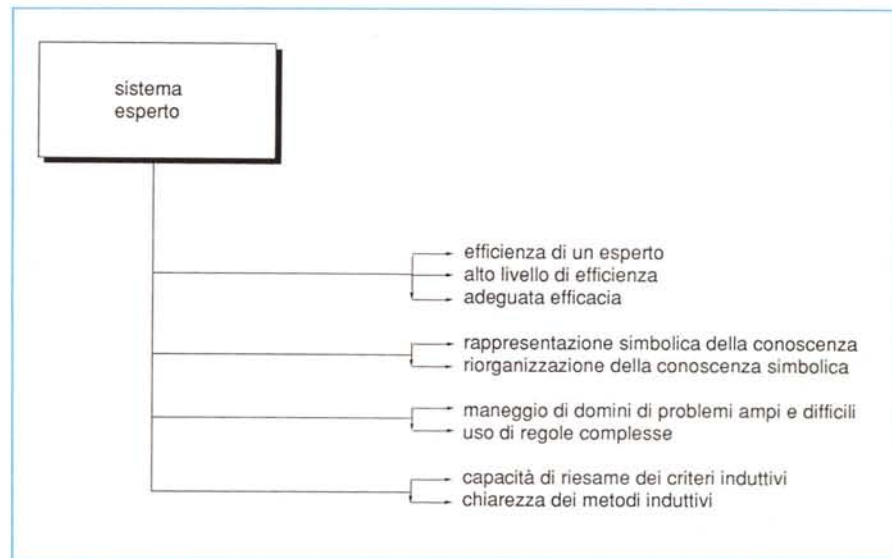


Figura b - Caratteristiche specifiche di un sistema esperto.

usati) che costituisce il punto di forza e il cardine dello schema risolutivo del sistema esperto stesso. Ancora gli esperti umani, spesso inconsciamente (si pensi al lavoro di un medico e in particolare di un pediatra), sono costretti a una base di conoscenza e a trasformarla in dati da loro stessi interpretabili. Questa capacità di «riformulazione» rappresenta talora una necessità irrinunciabile, anche solo per rendere gli stessi S.E. simili ai loro corrispondenti umani (ve lo immaginate un contadino di Bisaccia, paesino sperduto tra le montagne di Avellino, che dialoga per telefono con un S.E. che gli deve diagnosticare un enfisema polmonare). Purtroppo molti sistemi esperti mancano di questa possibilità e sebbene efficientissimi in mano a persone superalleniate non contribuiranno certo, con la loro ermeticità, a rendere gli S.E. popolari.

Potenza

Un sistema esperto deve essere potente; proprio perché esso opera in un ampio ventaglio di possibilità, ancorché limitate ad un singolo argomento, ha necessità di essere, come dicevamo precedentemente, efficiente e veloce, pur restando mi si perdoni il termine, «giudizioso». Per tale motivo le regole in un sistema esperto sono necessariamente complicate, sia per la loro effettiva complessità, sia per il loro puro e semplice numero.

A questo punto è necessario eseguire un piccolo distinguo nell'area di inte-

resse del nostro discorso. Un sistema esperto lavora su un'area di studio diversa da quella di un puro studioso di Intelligenza Artificiale; i primi lavorano su domini del mondo reale, mentre gli A.isti esplicano la loro attività sul campo del dominio dei giochi (o dominio dell'immaginazione). Nel primo caso, il dominio del mondo reale, il tecnico incaricato di risolvere il problema applica i dati in suo possesso a problemi pratici e produce soluzioni che sono efficaci all'atto pratico. Nel dominio dei giochi, invece, il problema subisce generalmente una grossa semplificazione, e quasi sempre è un non realistico adattamento di alcuni problemi complessi del mondo reale. In questo caso il risolutore maneggia dati artificiali semplificati per rendere il problema più facile e produrre soluzioni che hanno interesse solo teorico.

Un esempio di applicazione nella vita reale è rappresentato, ad esempio, da una causa civile (un esempio del genere è esposto e risolto in Waterman e Peterson, «*Models of Legal Decision Making*», Rand Report, Rand Corporation, 1981). I dati relativi al problema legale da risolvere devono includere i fatti accaduti e descritti, i resoconti dei testimoni, i referti medici, la corrispondenza tra gli avvocati, e tutta la documentazione legale attinente al caso stesso.

Un esempio di domini di giochi — «Shangai», un gioco ben noto che abbiamo visto girare in una magnifica realizzazione anche sul Macintosh. Si tratta, come tutti sapranno, di un antico

gioco di combinazione di pezzi del domino; inutile dilungarsi sull'argomento. Un programma che produce piani per la soluzione del problema, non può considerarsi un sistema esperto.

I ricercatori di A.I. impegnati nei sistemi esperti cercano di evitare il dominio dei giochi pressoché completamente; si acquista, in caso contrario, una mentalità risolutiva che può falsare poi l'approccio alla soluzione del problema reale. Tanto per intenderci una massiccia semplificazione del problema può portare a dare risposte non proporzionate all'effettiva problematica iniziale. Al contrario, i metodi necessari per rappresentare e organizzare la conoscenza e per applicare tecniche di risoluzione dei problemi a questa stessa conoscenza sono sovente ostacolati dalla mole e dalla complessità della cosiddetta «area di ricerca», il fascio di soluzioni possibili, intermedie e finali, del problema. Semplificando il problema, ovviamente, come nel campo dei giochi, il programma probabilmente sarà più efficiente, ma neppure un esperto di S.E. saprà davvero quanto vicina alla realtà sarà poi la risposta finale.

Autoconoscenza

Un sistema esperto possiede una conoscenza di base che può essere manipolata dalle sue strutture operative e dai suoi processi razionali, ancorché semplificati. Ad esempio, un sistema esperto organizzato attraverso una serie di regole può essere agevolmente analizzato attraverso la rappresentazione del suo motore inferenziale per evidenziare la sua tecnica di giungere alla soluzione del problema stesso. Ma nell'ambito di tali regole ce ne possono essere altre che controllano le prime testando continuamente la loro efficienza, consistenza e plausibilità nei confronti dei campi di conoscenza che il S.E. maneggia. Questo sistema di supervisione della conoscenza, che poi non è altro che la conoscenza della conoscenza, è chiamata in gergo A.I.: «metaconoscenza».

Molti sistemi esperti possiedono una struttura interna denominata *facility* destinata alla spiegazione. Si tratta di una struttura cognitiva che spiega come un S.E. raggiunge le sue conclusioni e le sue risposte; ovviamente la maggior parte di queste strutture (che possono essere anche più di una per ogni programma) sono strutturate attraverso motori inferenziali.

La presenza di tali strutture, che sono visibili anche all'utente finale, è importante, sia dal punto di vista strutturale che fisico e psicologico. L'autoconoscenza è importante per una serie di

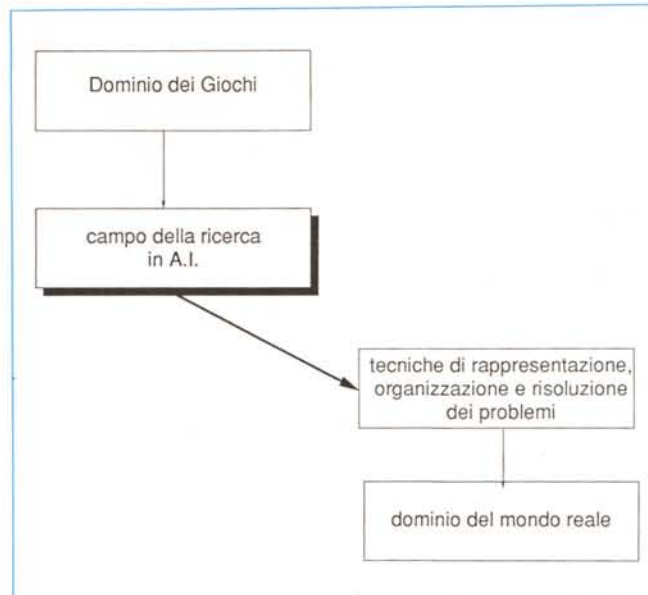


Figura c
Le correlazioni tra il mondo dei giochi e il dominio reale.

ragioni diverse: l'utente finale ha sempre sotto gli occhi le fasi che lo hanno portato ai risultati, e tende ad avere sempre maggiore confidenza col sistema e fiducia in esso; lo sviluppo del sistema è più semplice e il debug più rapido e intuitivo; tutte le operazioni logiche sono chiare all'operatore ed è più facile predire e testare l'effetto di un cambiamento nelle variabili concorrenti al problema stesso.

Si tratta di una struttura che non sempre è stata adeguatamente utilizzata; il primo passo per ampliare tale tecnica è di rendere nel sistema esperto il campo della metaconoscenza esplicito e separato dal corpo principale del programma. Un esempio pratico di metaconoscenza, applicato ad un esempio reale potrebbe essere:

— se si applicano più regole per la soluzione di un problema, usare regole testate da esperti più che da gente sconosciuta.

Sembra ovvio, no? Ma solo alla mente di un umano abituata alla metaconoscenza che la applica senza rendersene conto! Ma considerate un poco quanto vi dico adesso: nel caso di una malattia un ricercatore sconosciuto ha sperimentato una terapia efficace; ovviamente anche se si tratta di uno sconosciuto lo specialista in mancanza d'altro, pur di salvare il paziente adotta le tecniche dello sconosciuto (è la tecnica di «Purtosto che la morte!», che rappresenta quello che nella passata puntata avevamo definito senso comune). Viceversa, in caso di diverse tecniche adottate da sperimentatori differenti lo specialista si adegua a quelle più sperimentate o a quelle utilizzate da persone più esperte (ancora una volta applicazione del senso comune). Tutto questo il S.E. non lo sa;

ecco che bisogna fornirgli regole di metaconoscenza perché possa applicarle alla conoscenza stessa.

Il problema dell'errore

Per chiudere la puntata parliamo adesso di un altro importante aspetto in cui sistemi esperti e programmi convenzionali differiscono. Mentre i secondi (immaginate uno spreadsheet o un word processor) sono costruiti per dare la risposta o per reagire nella maniera giusta in ogni momento, i sistemi esperti sono costruiti per dare per quanto possibile sempre la risposta giusta; ciò non toglie che talvolta possa capitare di sentirsi dare da un S.E. una risposta sbagliata (capita nelle migliori famiglie di medici e ingegneri).

Ma non è proprio così, come fa acutamente notare McDermott in J. McDermott, «*R1's Formative years*», A.I. Magazine, vol. 2, #2 1981; anche i programmi convenzionali commettono errori, ma si tratta di errori più subdoli in quanto, proprio perché dotati di una operatività numerica indiscutibilmente esatta, se errore c'è esso sta nella impostazione e nelle premesse del programma e non nel codice. Se errore invece si verifica in un sistema esperto, esso risiede probabilmente in una non prevista variante del problema; essi cioè, hanno la incomparabile possibilità di imparare dai loro errori e di modificare il codice ogni volta che sia possibile per includere nuove varianti del problema, finora non previste o previste in parte.

Perciò, come dico anche in altra parte della rivista, a ognuno il suo! A risentirci la prossima volta per cercare di capire a cosa è applicabile con successo un sistema esperto.

Lo cercavo veloce e sicuro...

PHOTO: A. MEDRI ART: C. LAMPARIELLO



...e poi ho trovato LUI

COMPUTER
HSP
COMPUTER



AT da 1.999.000

MhZ 512 Kb ESP 4 Mb FD 1.2

386 da 2.310.000

20/25 MhZ 1 ESP 8 Mb FD 1.2

MEMORIE ROTANTI

FDD 144 Mb TEAC	190.000
HDD 20 Mb SEAGATE	349.000
HDD 40 Mb 28 MS SEAGATE	650.000

SCHEDE GRAFICHE

SUPER EGA 640 480	290.000
VGA 800 600 8 bit 256 K ESP	420.000
ULTRA VGA 1024 768 16 bit	480.000
EPSON LX 800	410.000
EPSON LQ 500	592.000

CITIZEN 180 E	310.000
---------------	---------

SWIFT 242

650.000

INFORMATICA D'AUTORE



INFO.SIST.

CONCESSIONARIA

LINEA

14" B/N HERC/CGA BASC	190.000
14" SUPER EGA COL. 640x400 DP 031	650.000
MULTISYNC VGA 1024x768	850.000
MONITOR NEC 2A,3D	da 999.000

VARIE

MOUSE GENIUS	da 60.000
PLOTTER ROLAND A3/A4	1.600.000
MODEM 300/1200 COMP. HAYES	138.000
MODEM 300/1200/2400 COMP. HAYES	250.000

NEC LINE

P 2200 24 AGHI	590.000
P 6 PLUS	1.100.000
P 7 PLUS	1.490.000