

Sistema esperto nelle biblioteche

Il Servizio Bibliotecario Nazionale si avvale di un sistema esperto per un servizio direttamente aperto al pubblico che permette al lettore di scegliere fra una vasta gamma di prestazioni qualificate.

La tecnologia del sistema esperto non rappresenta ormai in linea di massima una novità; tuttavia le applicazioni operative risultano ancora problematiche, o per lo meno non hanno raggiunto quella diffusione che sarebbe lecito aspettarsi. In buona parte la ragione risiede nel ritardo con cui si sono mosse le case di software nostrane e nella difficoltà oggettiva ad adattare alle esigenze dell'utente italiano gli ambienti di sviluppo di importazione. Per questo motivo la scelta che ben pochi produttori hanno tentato, cioè la realizzazione in proprio di «shell» e di programmi «esperti», senza limitarsi alla semplice commercializzazione di prodotti esteri, sta rivelandosi concorrenziale.

Se si prescinde dalle implementazioni in Prolog, più ancora che in Lisp, di sistemi esperti in genere piuttosto limitati nelle funzionalità e nelle prestazioni, implementazioni spesso circoscritte ad ambiti universitari e di ricerca, in Italia la produzione autonoma e originale è stata minima. L'interesse dell'applicazione illustrata, ovvero l'utilizzo del sistema esperto per rappresentare l'esperienza e la competenza del bibliotecario nel rapporto quotidiano con l'utente della biblioteca pubblica, sta anche nella genesi tutta particolare degli strumenti tecnici di base.

Il guscio «Ergo»

La struttura dello shell Ergo (questa volta, non un acronimo, ma una elementare paroletta latina...) è stata realizzata in C su un'idea di fondo che sposa l'impostazione formale della logica deduttiva all'atteggiamento prosaico e concreto della tradizione applicativa. L'ipotesi di lavoro è che non occorra uno strumento eccessivamente sofisticato per ottenere i risultati più significativi; troppi optional finiscono anzi per sconcertare l'utente e non facilitano il già complesso rapporto tra ingegnere della conoscenza e esperto

umano. Non a caso chi ha utilizzato shell precostituiti si è in primo luogo occupato di ritagliare la parte da utilizzare veramente, operando, per così dire, in levare. Qui, più economicamente, si è lavorato...in mettere, dotando cioè il nucleo del programma solo di quelle caratteristiche funzionali al lavoro sul campo. Le componenti del sistema sono dunque una base di conoscenza a regole e un motore inferenziale backward-forward corredato di un'interfaccia utente meno scoraggiante possibile. La base di conoscenza è strutturata come un normale data-base gestionale ed è definita tramite il Dbms Informix, in ambiente Unix o Dos. La regola appare così, in fase di gestione della base, come una classica entità relazionale; tra i suoi attributi, vi sono i codici degli asserti di premessa e di conclusione, che rinviano alla apposita tabella di convalida. Rinunciando a raffinati meccanismi di formulazione predicativa, si è ricorso a banali codici (numerici, perché no?) e alle rispettive decodifiche in lingua: il frasario noto al sistema è praticamente tutto nel master degli asserti definiti, ampliabile a piacere. Ciò permette la massima rapidità nella formulazione delle regole, e in sostanza della conoscenza del sistema, evitando nel contempo ogni ambiguità indesiderata. Esorcizzata quindi la knowledge-base, vediamo l'ossatura del motore inferenziale, elemento cruciale di ogni sistema esperto. Il C, linguaggio molto veloce ed essenziale, ha permesso un ricorso disinvolto a strutture di dati e a processi ricorsivi, garantendo un'impostazione generalizzata e formalmente corretta. Ogni luogo della rete logica definita dall'insieme di regole è omologo a qualsiasi altro: ogni punto può divenire l'obiettivo particolare da cui ricostruire e ripercorrere l'albero deduttivo che virtualmente vi fa capo. Una asserzione dichiarata dall'operatore, ricavata da un archivio esterno oppure dedotta in base alle regole di produzione, innesca

una rivisitazione dell'intera rete, alla ricerca di ulteriori nuove conseguenze. La ricorsione si arresta di fronte ad anelli chiusi, nel senso che l'albero deduttivo può anche essere un grafo complesso; in questo caso il carattere «indeterministico» dell'esame delle dipendenze logiche è ancora più marcato e diventa difficile (cosa del resto quasi mai utile) intervenire sull'ordine di esecuzione con scorciatoie o differenti priorità. Pur essendo consentito definire doppie implicazioni e ragionamenti circolari, per sveltire il lavoro di costruzione della base e in ultima analisi lo stesso processo deduttivo si è preferito introdurre altri costrutti logici, come l'equivalenza (per dichiarare varie frasi tutte con lo stesso «significato», ovvero con lo stesso valore di verità) e l'alternativa esclusiva (non appena un asserto risulta vero, tutti gli altri della stessa lista sono considerati falsi). Il motore inferenziale opera inoltre mediante *modus tollens*, per cui dalla falsità della conclusione si ricava la falsità della premessa. Tutto questo non comporta grosse rigidità rispetto al ragionamento ordinario e il comportamento del programma di fronte alle regole che gli vengono sottoposte è risultato accettabile. Anche la differenza tra l'operare in backward, cioè orientato al fine prescelto, e in forward, «in avanti» su tutto l'insieme di regole, ha un contraltare pratico: il backward è la modalità assunta dal sistema quando pone le domande ed esamina le lacune in vista dell'obiettivo; il forward può utilizzarsi quando s'intende verificare in parallelo tutta la conoscenza immessa, prescindendo dalle *potature* già effettuate.

L'interfaccia utente

Mentre la manipolazione della base di conoscenza si avvale essenzialmente di procedure Informix non dissimili da ordinarie funzioni di interrogazione e gestione di archivi, l'interfaccia di col-

