

PROTOTIPIZZAZIONE ED EXPERT SYSTEM SHELL: UN ESEMPIO D'USO DI XI

Silvano Marioni

Banca Solari & Blum S.A.

Lugano

RIASSUNTO

L'articolo illustra l'impiego di un expert system shell come strumento per costruire e sperimentare un modello che sarà poi sviluppato con un linguaggio di programmazione tradizionale. Viene proposto un caso pratico nel settore della tariffazione delle commissioni di borsa sviluppato con l'impiego dell'expert system shell Xi.

ABSTRACT

This paper presents the applications of an expert system shell for building and testing a model before developing and programming with a programming language. The case study is showing an application using the expert system shell Xi in the area of stock exchange regulations.

1. INTRODUZIONE

Le continue modifiche, tipiche dei processi di manutenzione del software, richiedono una costante evoluzione dei programmi applicativi e spesso l'introduzione di ulteriori cambiamenti può portare alla necessità di revisione di una intera applicazione.

Negli ultimi anni si è investito molto nello studio e progettazione di linguaggi che permettono di ottenere il risultato di un problema descrivendo le sue regole senza dover necessariamente indicare il loro ordine di applicazione (fig. 1). Uno dei risultati più importanti a cui porta l'adozione di linguaggi di questo tipo è una maggiore facilità sia nei processi di sviluppo che di manutenzione del software.

Il processo di sviluppo di un programma comporta in una prima fase la definizione di un modello che comprenda la

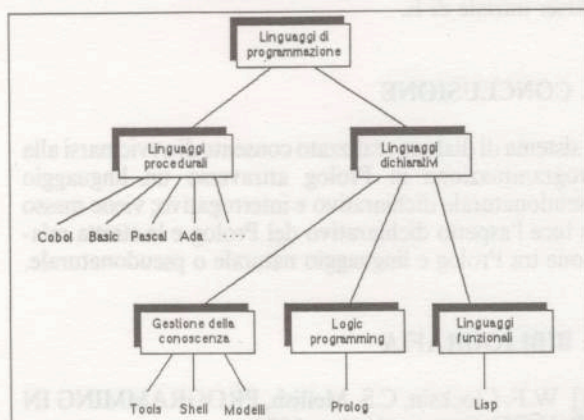


Fig. 1 Linguaggi di programmazione procedurali e dichiarativi

conoscenza del problema.

Dopo la formalizzazione del modello è possibile affrontare la fase che porta alla creazione dell'algoritmo. Solo a questo punto si è in grado di iniziare a scrivere le istruzioni in codice.

Ma che cosa succede se, dopo aver ideato e programmato l'algoritmo, scopriamo che la nostra conoscenza del problema non era completa?

Non è facile aggiungere i pezzi di conoscenza mancante a quella che abbiamo già congelata nell'algoritmo perché ci si scontra con le difficoltà che un linguaggio di programmazione procedurale, cioè sequenziale, ha nell'elaborare una situazione complessa di tipo dichiarativo. Questi inconvenienti si incontrano con sempre maggior frequenza perché la carenza di comunicazione tra utente e programmatore è aumentata con il crescere di complessità

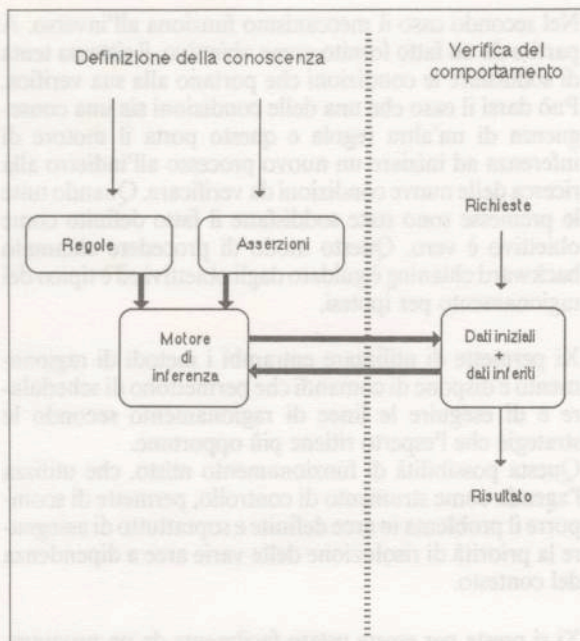


Fig. 2 Definizione della conoscenza con un Expert System Shell

dei sistemi informativi. L'utente che desidera un programma per risolvere un suo problema ha spesso una visione limitata e parziale perchè non ha più il possesso diretto dei dati e non è in grado di descrivere un modello adeguato del suo problema. Il programmatore d'altro canto non riesce a comprendere il problema se non dalla descrizione data dall'utente. Se a tutto questo aggiungiamo le ambiguità del linguaggio quando si affrontano problemi di una certa complessità, non è difficile immaginare che gli errori nella comunicazione della conoscenza possono incidere in modo determinante nei tempi e nella qualità del lavoro di programmazione.

Per ovviare a questi inconvenienti la soluzione più efficace è quella di provare in anticipo il problema con strumenti che permettano di verificare il modello in un modo completo e che rendano semplici gli eventuali processi di revisione della conoscenza.

Tra gli strumenti che oggi permettono di applicare la metodologia della prototipizzazione, i linguaggi per la gestione della conoscenza, e tipicamente gli expert system shell, non richiedono la definizione di un algoritmo a priori, ma permettono una definizione estremamente modulare della conoscenza come regole separate una dall'altra (fig. 2).

Questa caratteristica permette di iniziare il processo di costruzione e di verifica aggiungendo nuove regole quando servono e rendendo possibile la definizione dell'algoritmo in modo incrementale, senza avere ancora definito il modello.

Un expert system shell diventa uno strumento di indagine del modello che sarà sviluppato in seguito in un linguaggio di programmazione tradizionale.

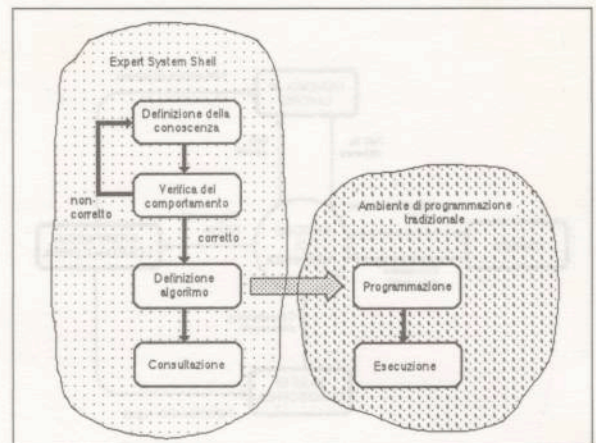


Fig. 3 Regole definite nel sistema esperto e nel programma tradizionale

Si può porre a questo punto l'obiezione che gran parte degli algoritmi di un programma riguardano le parti più tecniche, quali la gestione delle videte o l'rganizzazione e i metodi di ricerca dei dati.

L'expert system shell non si pone come un metodo per costruire in precedenza tutti gli algoritmi di un programma, ma solo quelli di una discreta complessità e che devono essere verificati con la collaborazione di un esperto.

Alla fine del lavoro di progettazione dell'algoritmo uno degli effetti collaterali da non trascurare è quello di avere costruito un vero e proprio sistema esperto nel settore di applicazione dell'algoritmo. La conoscenza del problema fornita dall'esperto e operante nel programma tradizionale, è presente anche in un nuovo modo nel sistema esperto a cui si può fare riferimento per modificare e sperimentare l'algoritmo o per utilizzarla come fonte di documentazione (fig. 3).

2. L'EXPERT SYSTEM SHELL XI

Xi è un ambiente di programmazione della conoscenza funzionante su un personal computer che permette la costruzione e la consultazione di un sistema esperto. È la versione commerciale e migliorata di Props, un sistema sviluppato presso l'Imperial Cancer Research Fund Lab., uno dei maggiori centri di ricerca inglesi. Xi è composto da uno shell e da una serie di strumenti per facilitare la memorizzazione e la gestione della conoscenza. Sviluppato e commercializzato attualmente dalla ditta Expertech, è scritto in Prolog-2 e funziona su un P.C. con almeno 380 Kb di memoria. Il linguaggio di programmazione di Xi permette la composizione di regole e di asserzioni memorizzate in forma esplicita nel calcolatore con una sintassi molto vicina alla lingua inglese. È possibile una gestione dell'input e output che comprende la generazione automatica di menu per l'entrata dati, la gestione dei report di output sia su video che su printer, l'accesso a file su disco. Il motore di inferenza permette di adottare sia la tecnica del forward chaining che quella del backward in modo da permettere un'ottimizzazione nella consultazione della base di conoscenza.

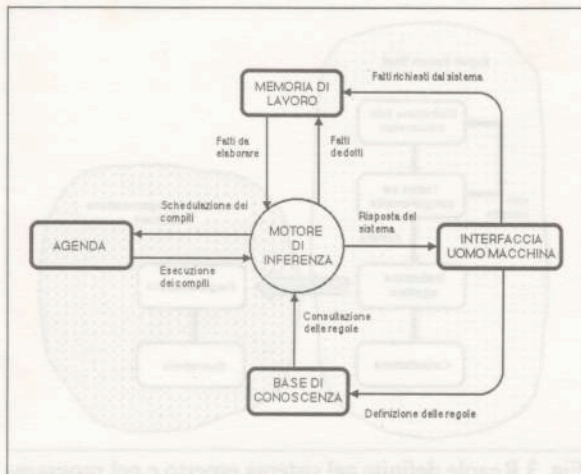


Fig. 4 Architettura dell'Expert System Shell Xi

Progettato volutamente come strumento utilizzabile da persone non esperte di informatica può essere usato in due modi a seconda della necessità di impiego. Il modo "menu", si presta per il lavoro di consultazione della base della conoscenza e permette un utilizzo estremamente semplice del sistema esperto utilizzando i menu come strumento di dialogo per fornire tutti i dati richiesti. Il modo "dialogo" è un modo più sofisticato di fare consultazione ma soprattutto è il modo per definire e introdurre la conoscenza e costruire il sistema esperto.

Xi permette di costruire sistemi esperti utilizzando il metodo delle regole di produzione. L'architettura del sistema esperto è composta da una parte statica, la conoscenza memorizzata, e una dinamica, le situazioni che si vengono a creare nel contesto di una consultazione (fig. 4).

La parte statica è formata da un insieme di conoscenze su un particolare dominio che è espresso sotto forma di regole e fatti viene chiamato comunemente "base di conoscenza".

La parte dinamica è formata da una memoria di lavoro dove vengono poste tutte le conclusioni derivate dalle regole per essere riutilizzate, in fasi successive, fino al raggiungimento dell'obiettivo: la gestione di questa parte della dinamica è operata dal motore inferenziale.

Xi dispone inoltre di una ulteriore parte chiamata "agenda", che sovraintende al controllo delle linee di ragionamento.

Vale forse la pena di ribadire il modo in cui le regole e i fatti della base di conoscenza sono utilizzati in un sistema esperto, per poter dare una giusta collocazione a Xi. Esistono in effetti due strategie che corrispondono a due modi di ragionamento. Nel primo caso si possono fornire a priori tutti i fatti di cui si è a conoscenza. Se alcuni di questi fatti fanno rendere vera una regola questa aggiunge alla memoria di lavoro la sua conclusione e questo può far ricominciare un nuovo processo di deduzione fino a quando non è più possibile dedurre nuovi fatti. Questo modo di procedere chiamato *forward chaining* è guidato dai fatti ed è tipico del ragionamento deduttivo.

Nel secondo caso il meccanismo funziona all'inverso. A partire da un fatto fornito come obiettivo, il sistema tenta di soddisfare le condizioni che portano alla sua verifica. Può darsi il caso che una delle condizioni sia una conseguenza di un'altra regola e questo porta il motore di inferenza ad iniziare un nuovo processo all'indietro alla ricerca delle nuove condizioni da verificare. Quando tutte le premesse sono state soddisfatte il fatto definito come obiettivo è vero. Questo modo di procedere chiamato *backward chaining* è guidato dagli obiettivi ed è tipico del ragionamento per ipotesi.

Xi permette di utilizzare entrambi i metodi di ragionamento e dispone di comandi che permettono di schedulare e di eseguire le linee di ragionamento secondo le strategie che l'esperto ritiene più opportune.

Questa possibilità di funzionamento misto, che utilizza l'agenda come strumento di controllo, permette di scomporre il problema in aree definite e soprattutto di assegnare la priorità di risoluzione delle varie aree a dipendenza del contesto.

Xi si presta per essere usato facilmente da un programmatore che descrive le regole e le asserzioni per risolvere il problema senza tener conto dei problemi di controllo ma dedicandosi solo al loro contenuto logico.

Contemporaneamente anche l'utente come esperto del problema è in grado di verificare tutti i possibili comportamenti, isolando i malfunzionamenti, e permettendo al programmatore di modificare in modo rapido il modello del problema.

3. UN ESEMPIO DI USO DI XI

La Banca Solari & Blum ha utilizzato l'expert system shell Xi come strumento di prototipizzazione in occasione dell'introduzione della nuova Convenzione di borsa svizzera alla fine del 1985.

Il modello della nuova convenzione è chiaramente definito in un documento del Vereinigung Schweizerischer Effektenbörsen che specifica i modi e i tipi di operazioni di borsa e fissa i tassi da applicare all'operazione. Per le operazioni concluse in Svizzera sono stabiliti dei tassi in funzione del tipo del titolo e del valore di borsa della transazione (Art. 9.1 - 9.6). A questo si deve aggiungere per le operazioni concluse all'estero e fino al controvalore di Fr. 250.000 un nuovo tipo di logica che fa dipendere i tassi dalla nazione (Art. 10.1 - 10.2). Quando il controvalore supera Fr. 250.000 si riprende la regola delle operazioni concluse in Svizzera con l'aggiunta delle commissioni estere effettive (Art. 11). Se l'operazione conclusa all'estero è al netto di commissione estera si applicano le regole per le operazioni concluse in Svizzera (Art. 12). La convenzione dà inoltre indicazioni precise per classificare i tipi di titolo in funzione della nazionalità o della divisa o di altre caratteristiche del titolo.

Questo non vuol essere il riassunto della convenzione di

L'identificatore *transazione* per essere verificato necessita una linea di ragionamento differente. Il motore di inferenza trova che gli unici punti dove può essere valutato sono delle conseguenze di altre regole e quindi inizia un processo di backward chaining per soddisfarle. Le regole interessate sono definite qui di seguito

```
if borsa is svizzera
then transazione is svizzera
```

```
if borsa is not svizzera
and al netto is no
and importo <= 250000
then transazione is estera
```

```
if borsa is not svizzera
and al netto is no
and importo > 250000
then transazione is estera
```

```
if borsa is not svizzera
and al netto is si
then transazione is svizzera
```

Anche in questo caso gli identificatori *importo*, *al netto*, e *borsa* sono riconosciuti come dati forniti dall'esterno e quindi vengono impostati ponendo delle domande all'utente.

```
question importo
Imposta l'importo in franchi svizzeri
range 0 to 8.000.000
```

```
question al netto
L'operazione è al netto della commissione estera
one of
si,
no
```

```
question borsa
In quale borsa è trattato il titolo
one of
svizzera,
usa,
giappone,
inghilterra,
. . . .
. . . .
```

Una volta definito l'identificatore *articolo* il motore di inferenza procede con dei processi analoghi alla definizione di *tranche importo* e contemporaneamente alla visualizzazione del risultato.

Le istruzioni appena viste possono servire per dare un'idea a livello più tecnico del linguaggio usato nell'expert system shell Xi.

Questo è solo un campione dei circa 200 elementi, quali regole, fatti, e domande, che compongono la base di conoscenza e che sono state ottenute come risultato del lavoro comune di impostazione e di revisione svolto dal programmatore e dell'esperto di borsa.

4. UN'ESPERIENZA PRATICA

Un punto delicato nel processo di formalizzazione dell'algoritmo è dato dalla necessità di interfacciare le nuove logiche con il sistema informativo esistente. È naturale affrontare il problema con una forma mentale che porta a vedere le nuove modifiche in funzione dei modelli di conoscenza presenti nell'azienda. Le nostre aspettative ci fanno assumere come acquisito ogni concetto che ha un alto grado di somiglianza con i concetti che ci sono familiari. Spesso si trascurava di approfondire il concetto nella lettura delle nuove specifiche e si rischia di non rendersi conto delle differenze che, anche se appaiono in modo sfumato, possono essere fondamentali. Un esempio di questa possibilità di errore può servire come chiarimento; quando nella convenzione di borsa si parla di obbligazioni in franchi svizzeri si è portati a pensare alla divisa in cui è trattata in borsa l'obbligazione. Una lettura più attenta della convenzione specifica che si deve considerare la divisa in cui il titolo sarà rimborsato alla scadenza. Il fatto che nella maggior parte dei casi la divisa di trattamento e la divisa di rimborso sono le stesse facilmente può trarre in inganno e portare a considerare solo la divisa legata al titolo.

Questo tipo di errore "di disattenzione" può essere evitato utilizzando in prima stesura nel sistema esperto gli stessi testi della convenzione di borsa e sicuramente, riprendendo la logica in fase di creazione dell'algoritmo, sarà più facile prestare attenzione alle differenze.

Questa che può sembrare un'osservazione marginale è invece un punto che merita di essere approfondito nell'ottica di utilizzo di un expert system shell. Il lavoro di analisi che tradizionalmente viene fatto prima di iniziare a scrivere il programma è di tipo statico e richiede il passaggio alla fase di programmazione per ottenere una verifica. L'expert system shell offre delle possibilità in più. La logica può essere ricopiata direttamente dalle specifiche alla base di conoscenza, ma quello che più conta può essere anche verificata immediatamente in modo dinamico facendo funzionare il prototipo di sistema esperto fino ad ottenere il comportamento desiderato.

L'estrema semplicità con cui le disposizioni possono essere riprese dal testo e scritte nel linguaggio di un expert system shell è una delle grosse novità rispetto allo stile di programmazione tradizionale.

La conoscenza è espressa in forma esplicita. Se trascuriamo il fatto che alcune parole chiave sono espresse in lingua inglese possiamo notare che la leggibilità delle regole è molto vicina a quella del testo originale.

Questa caratteristica permette di utilizzare le regole per dare spiegazioni sulla linea di ragionamento che il sistema esperto sta seguendo. Possiamo chiedere spiegazioni sul "come" una certa conclusione è stata raggiunta e sul "perchè" una certa domanda viene posta. La risposta sarà data con la visualizzazione della regola o della serie di regole che in quel preciso contesto sono prese in considerazione dal motore di inferenza.

Le regole sono indipendenti una dall'altra, vengono impostate senza preoccuparsi del loro ordine di esecuzione e

possono essere corrette senza nessun pericolo di alterare la sequenza di esecuzione.

Non è necessario lo scrivere le regole in modo strutturato. Nella base di conoscenza le regole sono raggruppate secondo alcuni criteri logici ma questo viene fatto solo per rendere più chiara la lettura.

Alcuni expert system shell (Xi è uno di questi) hanno la possibilità di effettuare un controllo di coerenza logica delle regole nella base di conoscenza. Diventa possibile fornire le regole in modo frammentario e utilizzare il sistema come uno strumento di aiuto per stabilire le corrette relazioni che nei problemi di una certa complessità logica spesso restano nascoste.

5. CONCLUSIONE

I problemi per i quali è possibile definire un algoritmo di risoluzione non sono in genere considerati di competenza dei sistemi esperti.

Un expert system shell, come nuova tecnologia di programmazione, è preferito nei casi in cui la conoscenza è meno definita e dove la soluzione è ottenuta applicando delle regole di tipo euristico, perchè la struttura logica non è stata ancora chiarita e formalizzata. Sembra quindi una contraddizione risolvere un caso di programmazione tradizionale, dove tutto può essere definito con precisione, con uno strumento studiato soprattutto per affrontare problemi per i quali non sia noto con esattezza come arrivare ad una soluzione. In effetti la costruzione di un algoritmo passa attraverso momenti in cui la visione del problema non è sempre molto chiara.

In queste situazioni non esattamente formalizzate esiste un grosso spazio per l'applicazione di un expert system shell, che può essere di aiuto per passare dalla descrizione di un meccanismo logico all'algoritmo che lo riproduce, attraverso un'evoluzione di modelli successivi.

6. BIBLIOGRAFIA

- [1] S. Bandini, I SISTEMI ESPERTI, "Note di software", n. 23/24, Giugno 1984
- [2] C. Donalizio, PROGRAMMAZIONE LOGICA E PROLOG, "Informatica Oggi", Aprile 1986
- [3] C. Chiopris, P. Krauth, A. Markus, LINGUAGGIO PER (SISTEMI) ESPERTI, "Informatica Oggi", Maggio 1986
- [4] P. Szeredi, PROLOG THE NEW GENERATION LANGUAGE FOR APPLIED A.I., Atti del convegno "Intelligenza Artificiale: strumenti e applicazioni in ambiente Prolog", Padova, 1984
- [5] J. Fox, A SHORT ACCOUNT OF KNOWLEDGE ENGINEERING, Imperial Cancer Research Fund Lab., London, 1985

[6] P. Jackson, INTRODUCTION TO EXPERT SYSTEM, Addison-Wesley Publishing Co, Reading, Massachusetts, 1986

[7] D.A. Waterman, A GUIDE TO EXPERT SYSTEM, Addison-Wesley Publishing Co, Reading, Massachusetts, 1986

[8] R. Schank, L. Hunter, THE QUEST TO UNDERSTAND THINKING, "Byte", Aprile 1985

[9] A. d'Agapeyeff, THE EXERCISING AND SHARING OF DECISION KNOW-HOW IN BUSINESS, Expertech Ltd., Slough, 1985

[10] Xi REFERENCE MANUAL, Expertech Ltd., Slough, 1985

[11] Courtag-Konvention, Vereinigung Schweizerischer Effektenbörsen, Zürich, 1985