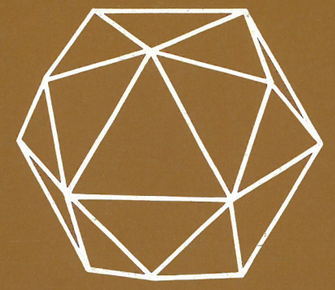
Spedizione in abbonamento postale a tariffa intera/Milano

Quaderni di informatica

Rassegna di tecnologia ed applicazioni degli elaboratori elettronici
Anno XIV - Numero 2 - 1988

33



CENTRO STUDI
INFORMATICA E AUTOMAZIONE

ARCHIVIO STORICO
ASSOCIAZIONE POZZO DI MIELE
FPDM-40
RQDI 131

Honeywell Bull

FPDM-40

Honeywell Bull

Quaderni di informatica

Rassegna di tecnologia ed applicazioni degli elaboratori elettronici
Anno XIV - Numero 2 - 1988

Sommario

Modelli connessionisti ed elaboratori neuronali

Non è ancora arrivata la quinta generazione di elaboratori, basata sui concetti della Intelligenza Artificiale "classica", che già si parla di una sesta generazione, caratterizzata dall'impiego di reti neuroniche artificiali. È questo, in effetti, un tema non nuovo ma che vede ora un rinnovato interesse a seguito di nuove idee nonchè dei progressi in varie discipline, dalla microelettronica alla neurologia. Su questo stimolante argomento l'autore presenta un conciso, ma chiaro e aggiornato panorama.

N. CESA BIANCHI

Un Sistema Esperto per la riconfigurazione di mainframe

La tecnica dei Sistemi Esperti trova sempre più numerosi e vari esempi di applicazione pratica. Nella realizzazione qui presentata, un Sistema Esperto viene usato per assistere l'operatore umano nella riconfigurazione di grandi sistemi di elaborazione in caso di guasto.

D. TOPPETA

Criteri di ricerca in grandi basi di dati multimodali

Come "navigare" in un grande Data Base multimodale, costituito da dati, testi, immagini, suoni, per raggiungere le informazioni richieste? A tale fine, viene qui proposta una architettura di accesso basata sul metodo detto di "filtering e browsing".

L. FELICIAN, D. FACCHIN

Tecniche di stampa per l'informatica individuale

La diffusione a macchia d'olio del Personal Computer ha stimolato lo sviluppo di tecniche atte a realizzare stampanti compatte e di basso costo, ma con ottime prestazioni sia come velocità che come qualità di stampa. L'autore fa una rassegna delle principali soluzioni tecnologiche, soffermandosi su quelle di maggior impiego e prospettive.

G.C. GATTI

Direttore Responsabile
Franco Filippazzi

Comitato di Redazione
A. Cicu, C. Falcetti, P. Lupo,
M. Nobile, G. Occhini
F. Sala

Redazione
Honeywell Bull
Centro Studi Informatica e Automazione
Via G.M. Vida, 11 - 20127 Milano

Stampa
tipolito Maggioni s.a.s.

Autorizzazione del Tribunale di Milano
n. 93 del 20 Marzo 1974

«Quaderni di Informatica» ospita
articoli e contributi di varia provenienza.
La responsabilità delle opinioni
espresse rimane agli Autori.

La riproduzione di articoli
della rivista, o di parte di essi,
è consentita solo citando la fonte
e l'autore.

Modelli connessionisti ed elaboratori neuronali

NICOLÒ CESA BIANCHI

Dipartimento di Scienze della Informazione
Università di Milano

1. INTRODUZIONE

Gli elaboratori neuronali o neurocomputer sono stati ormai identificati come la sesta generazione di sistemi per l'elaborazione delle informazioni. Essi rappresentano la realizzazione di un paradigma computazionale noto come *Connessionismo* che si caratterizza come il tentativo di riprodurre le capacità del cervello attraverso modelli formali che ne astraggano, in qualche modo, i dettagli strutturali e funzionali.

Questo tipo di approccio si differenzia radicalmente dalla metodologia tradizionalmente seguita dall'Intelligenza Artificiale (AI). Infatti, mentre nel paradigma dell'AI "classica" l'elaborazione consiste essenzialmente in un processo di manipolazione sequenziale di simboli, il Connessionismo propone un modello di computazione distribuita nel quale l'elaborazione è frutto di un processo collettivo ed altamente parallelo.

Uno dei tratti fondamentali del paradigma connessionista è che esso costituisce un naturale contesto per lo studio dell'apprendimento. In effetti, il Connessionismo "moderno" sviluppatosi in questi ultimi anni, trovò una prima naturale collocazione nell'ambito della psicologia cognitiva che lo utilizzò con successo proprio nelle teorie dell'apprendimento, laddove i precedenti approcci, mutuati dall'AI "classica", avevano ottenuto pochi risultati. Nell'ambito della *computer science* la possibilità di acquisire conoscenza tramite esperienza è estremamente vantaggiosa per il trattamento di quei problemi – tipici nell'ambito del pattern recognition – che non si prestano facilmente ad essere risolti con gli usuali strumenti di programmazione ad alto livello. Inoltre, l'elevato parallelismo delle reti neuronali offre la possibilità di operare in tempo reale utilizzando processori relativamente lenti, così come il cervello risolve un problema di riconoscimento in un tempo pari a qualche centinaio di volte il tempo di commutazione dello stato di un neurone.

2. Cenni storici

Il prototipo degli attuali modelli connessionisti venne sviluppato nel 1943 da McCulloch e Pitts [1] coll'intento di proporre un modello dell'attività del sistema nervoso. In quel periodo, contemporaneamente allo sviluppo della Cibernetica, cominciò ad affermarsi la visione del cervello come sistema per l'elaborazione delle informazioni.

L'unità funzionale base del sistema nervoso è costituita dalla cellula nervosa o *neurone* schematizzata in figura 1; si ritiene che il cervello umano ne contenga circa cento miliardi. Il carattere fondamentale dell'attività neuronale è la capacità di ricevere, elaborare e trasmettere informazioni sotto forma di segnali elettrici ed impulsi chimici.

McCulloch e Pitts scoprirono come la produzione di impulsi da parte di un neurone potesse essere paragonata ad un calcolo logico sui segnali ricevuti in ingresso.

Basandosi su questa interpretazione, essi elaborarono un modello nel quale si assumeva che ad ogni neurone fossero associati uno stato di attività, durante il quale esso emetteva un impulso, e uno stato di riposo durante il quale non accadeva nulla.

La transizione di stato avveniva, in modo deterministico e contemporaneamente per tutti i neuroni, secondo un calcolo booleano sugli impulsi provenienti dagli altri neuroni, assegnando convenzionalmente lo zero logico allo stato di non attività e l'uno logico allo stato di attività. Al di là delle enormi semplificazioni rispetto alle reali funzionalità delle reti nervose, il modello di McCulloch e Pitts ebbe vasta risonanza dando anche un rilevante contributo alla nascente teoria degli automi.

Un altro importante passo avanti venne compiuto circa sei anni più tardi da D.O. Hebb [2] che individuò la relazione fra l'apprendimento e un processo di modifica strutturale del cervello, processo che si focalizzava soprattutto nelle *sinapsi,* i siti di comunicazione fra un neurone e l'altro.

3. Il perceptrone

Ma soltanto nel 1962 queste idee vennero riunite in un modello di calcolo formale, il *perceptrone,* che era in grado di apprendere a calcola-

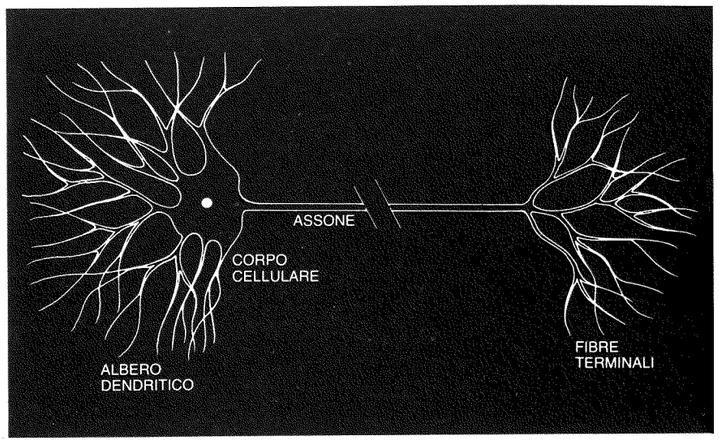


Fig. 1 - Schema di una cellula nervosa (Neurone). Gli impulsi provenienti da altri neuroni vengono ricevuti dall'albero dendritico, elaborati e trasmessi attraverso l'assone ad altre parti del sistema nervoso.

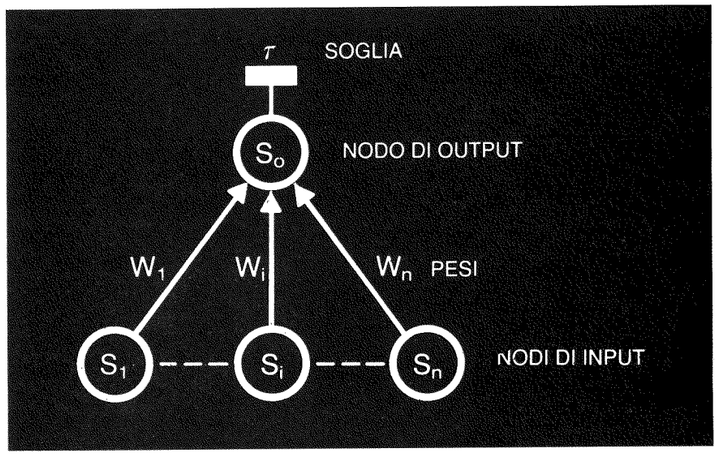


Fig. 2 - Il "Perceptrone" di Rosenblatt.

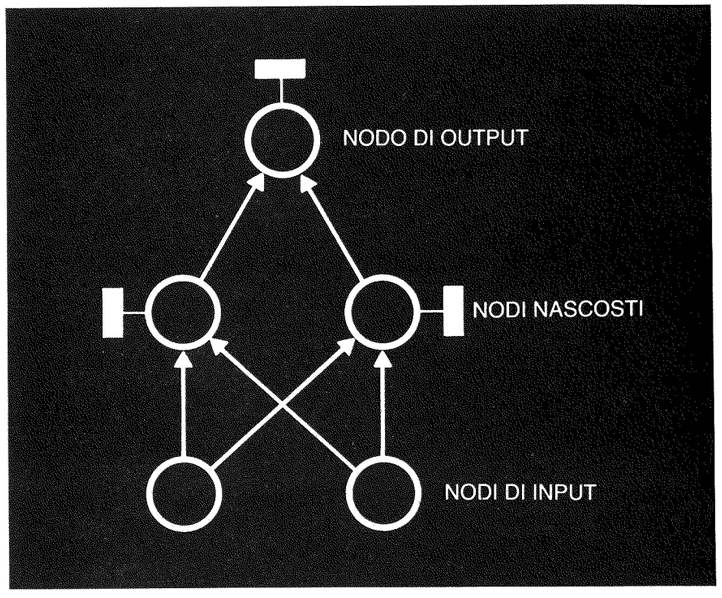


Fig. 3 - Un esempio di Perceptrone generalizzato a due livelli.

re delle funzioni mediante la presentazione di esempi costituiti da coppie argomento-valore appartenenti alla funzione stessa.

Il perceptrone ideato da Rosenblatt [3] (illustrato in figura 2) è composto da un insieme di *n nodi di input* collegati tramite *n* archi diretti ad un *nodo di output.*

Ogni nodo ha uno stato s che può assumere i valori di 0 e 1. La computazione si svolge assegnando un valore arbitrario allo stato di ciascuno degli n nodi di input; detta $\underline{s} = (s_1, \ldots, s_n)$ tale n-upla di valori di ingresso, l'output della computazione è rappresentato dallo stato s_0 del nodo di output che viene calcolato tramite la seguente *funzione di attivazione:*

$$s_0 = \begin{cases} 1 & \text{se } x_0 \geq \text{o} \\ 0 & \text{altrimenti;} \end{cases} \qquad (1)$$

dove:

$$x_0 = \sum_{i=1}^{n} w_i \, s_i - \tau \qquad (2)$$

Il perceptrone realizza quindi una particolare funzione booleana di n variabili. Il vettore $\underline{w} = (w_1, \ldots, w_n)$ e lo scalare τ, sono un insieme di numeri reali, detti rispettivamente *pesi* e *soglia,* che determinano, nel modo descritto da (1) e (2), lo stato del nodo di output in dipendenza della n-upla di valori presenti in ingresso. In genere, si associa ciascun peso w_i all'arco che va dall'i-esimo nodo di input al nodo di output, mentre la soglia r è associata al nodo di output (cfr. figura 2).
La struttura del modello impone una restrizione sulla classe delle funzioni da esso computabili. Si può facilmente dimostrare che una funzione booleana è computabile con questo dispositivo solo se è linearmente separabile (l'OR esclusivo è l'unico esempio di funzione booleana a due variabili non linearmente separabile).
Partendo dal presupposto che modificando i parametri di un perceptrone (cioè i pesi e la soglia) si può alterare la funzione da esso computata, Rosenblatt elaborò una *regola di apprendimento,* cioè una legge di modifica di questi parametri guidata da esempi forniti dall'esterno. Essa corrisponde alla seguente legge iterativa:

$$w_{i,k+1} = w_{i,k} + \eta(d(\underline{s}) - s_0)s_i \quad (3)$$

dove $w_{i,k}$ rappresenta il peso w_i al k-esimo passo iterativo, $d(\underline{s})$ l'output richiesto con $\underline{s}$ in ingresso, s_0 l'output prodotto dal perceptrone con $\underline{s}$ in ingresso e η un coefficiente maggiore di zero. Un teorema di convergenza dimostrato da Rosenblatt garantisce, tramite questa procedura, l'apprendibilità di qualsiasi funzione booleana linearmente separabile.

In realtà, la classe delle funzioni computabili da un perceptrone può essere estesa fino a coprire l'intero insieme delle funzioni booleane n-arie considerando modelli a più strati (perceptrone generalizzato) come mostrato in figura 3. In questo tipo di rete esistono dei *nodi nascosti* che costituiscono dei livelli di elaborazione intermedi.

La computazione procede attivando successivamente uno strato di nodi dopo l'altro a partire dallo strato contiguo ai nodi di input. Purtroppo, l'applicazione della regola di Rosenblatt a questo modello non risulta praticamente fattibile. Infatti, essa richiederebbe che ad ogni nodo (esclusi, ovviamente, quelli di input) venisse fornita l'informazione riguardante il corretto stato da assumere rispetto ad ogni possibile configurazione presente sui propri ingressi. Per problemi non banali questo si rivela estremamente complesso in quanto equivale a costruire una codifica del problema che sia consistente con la struttura della rete, compito che vorremmo fosse svolto dalla regola di apprendimento soltanto sulla base della specifica di un'unica relazione ingresso/uscita (quella fra nodi di input e nodo di output).

4. L'algoritmo di retropropagazione

La limitatezza di questo primitivo modello di rete neuronale fornì a Minsky e Papert [4] una base sufficiente per muovere delle pesanti critiche riguardo al paradigma connessionista; critiche che, di fatto, sancirono la sua esclusione dal campo dell'Intelligenza Artificiale a vantaggio dei paradigmi di elaborazione simbolica di tipo logico funzionale (Prolog e LISP).

Questa situazione mutò quando — nel 1985 — Rumelhart, Williams e Hinton [5] elaborarono l'algoritmo di retropropagazione dell'errore, una regola di apprendimento per il perceptrone generalizzato. Essa corrisponde all'applicazione del metodo del gradiente per minimizzare una misura della discrepanza fra l'output ottenuto e l'output desiderato rispetto ad un dato insieme di pattern di input.

Consideriamo una funzione $C(\underline{w})$ che misura, per esempio, lo scarto quadratico $(s_p - d_p)^2$ fra lo stato s_p assunto dal nodo di output quando la rete riceve in ingresso il pattern p e lo stato desiderato d_p per lo stesso input. Tale scarto dipenderà dal vettore $\underline{w}$ dei pesi e delle soglie della rete. Per applicare la regola del gradiente è necessario calcolare la variazione di C rispetto alla variazione di ciascun elemento w_i, cioè:

$$\frac{\partial C(\underline{w})}{\partial w_i}, \quad \text{per } i = 1,2, \ldots, N \qquad (4)$$

e quindi procedere all'applicazione della legge iterativa:

$$w_{i,k+1} = w_{i,k} - \eta \frac{\partial C(\underline{w}_k)}{\partial w_{i,k}} \qquad (5)$$

dove $\eta \geq 0$ è un coefficiente che regola l'ampiezza del passo.

Il calcolo delle espressioni indicate nella (4) richiede la differenziabilità della funzione di attivazione. Per questo motivo, le reti che utilizzano questa regola di apprendimento sostituiscono la (1) con l'approssimazione differenziabile:

$$s_i = \frac{2}{1 + e^{-x_i}} \qquad (6)$$

dove:

$$x_i = \sum_j w_{ji} \, x_j - \tau_i \qquad (7)$$

calcola, per il nodo i, il contributo fornito da ogni nodo j del livello inferiore pesato rispetto a w_{ji}. L'algoritmo di retropropagazione calcola le derivate parziali rispetto a tutti i pesi a partire dagli archi che terminano sul nodo di output. Il calcolo procede in direzione delle connessioni che si dipartono dai nodi di input in modo da sfruttare i risultati intermedi prodotti dai calcoli svolti negli strati superiori.

I modelli di tipo analogo al perceptrone generalizzato trovano naturale applicazione nell'ambito dell'apprendimento di mapping fra stringhe binarie. L'algoritmo di Rumelhart, Hinton e Williams a causa della sua flessibilità e semplicità implementativa si sta affermando come uno standard soprattutto per quanto riguarda le applicazioni di tipo commerciale.

5. I modelli simmetrici

Eliminando il vincolo che impone al perceptrone generalizzato un'architettura rigorosamente stratificata, possiamo considerare una più ampia classe di reti dove è anche ammessa l'esistenza di cammini ciclici nel grafo di rappresentazione (cfr. figura 4). Tuttavia, la non linearità della funzione di attivazione, unita alle retroazioni indotte dai cicli, rende estremamente complessa la descrizione dinamica di questo tipo di reti.

L'imposizione della simmetria dei pesi, cioè la richiesta che per ogni peso w_{ij} sulla connessione dal nodo i al nodo j esista una connessione di verso opposto con peso $w_{ji} = w_{ij}$, semplifica in qualche modo le proprietà delle reti cicliche. Nel 1982, Hopfield [6] elaborò un modello simmetrico in cui risultava definibile una *funzione di Ljapunov* sugli stati della rete. In altri termini lo stato del sistema, rappresentato dalla stringa binaria $(s_1, \ldots, s_n)$ degli stati di ogni nodo, evolve verso dei punti di equilibrio corrispondenti a minimi relativi della funzione di Ljapunov e determinati dai parametri numerici (pesi e soglie) della rete stessa.

Forzando il sistema in una configurazione iniziale qualunque, esso terminerà il proprio moto nello stato di equilibrio più vicino; questo comportamento può essere sfruttato per realizzare una memoria autoassociativa, cioè un dispositivo in grado di riprodurre interamente un dato memorizzato qualora gliene venga presentata una copia distorta o incompleta. A questo scopo, Hopfield ha sviluppato una regola di apprendimento che permette di calcolare i valori dei pesi di una rete in modo tale che un dato insieme di vettori binari risulti associato ai suoi stati di equilibrio.

Nel 1984, Hinton, Sejnowski e Ackley [7], hanno proposto l'implementazione sul modello di Hopfield di un algoritmo di ottimizzazione stocastica noto come "simulated annealing", dando così origine al modello della *macchina di Boltzmann*. Questo ha aperto la strada all'applicazione dei modelli connessionisti nel campo dell'ottimizzazione combinatoria (OC). In sostanza, riformulando la funzione costo di un qualunque problema di OC (per esempio il noto problema del commesso viaggiatore) come una funzione di variabili binarie, diventa possibile approssimare la configurazione di costo minimo utilizzando il fatto che una rete simmetrica minimizza, nel corso del proprio moto, una funzione di Ljapunov determinata dai pesi e dalle soglie; regolando opportunamente questi parametri è possibile interpretare tale funzione come la funzione costo del problema di OC in esame. Una piccola rassegna di applicazioni della macchina di Boltzmann all'ottimizzazione combinatoria si trova nel lavoro di Korst e Aarts [8].

6. Architetture per i neurocomputer

I modelli computazionali costruiti secondo il paradigma connessionista sono caratterizzati, dal punto di vista strutturale, da un'architettura a "grana fine", cioè prevedono la cooperazione di un numero elevato di processori ciascuno dotato di risorse computazionali limitate. In generale, il contributo dato dal singolo processore sarà minimo e dotato di scarso significato se isolato dal contesto globale. Infatti, in una rete neuronale ciò che risulta determinante è l'interazione fra l'insieme dei processori durante tutto il processo di elaborazione. Per queste peculiarità, in riferimento ai modelli connessionisti si parla di *pattern processing* in contrapposizione al *symbol processing* eseguito dai modelli di calcolo più tradizionali.

I principali problemi che l'implementazione di modelli neuronali comporta riguardano essenzialmente la gestione di un grande numero di processori che comunicano in modo possibilmente asincrono e la presenza di un alto numero di connessioni.

Elaboratori paralleli che sembrano adatti a implementare reti neuronali, anche se non progettati esplicitamente per questo scopo, sono gli array cellulari. Un array cellulare è costituito da un insieme di processori elementari ciascuno dotato di una piccola memoria locale e disposti in una struttura regolare in cui ogni processore è connesso ai suoi vicini più prossimi (cfr. figura 5). Il modello di calcolo è di tipo SIMD (Single Instruction Multiple Data), cioè esiste un controller che ordina ad un sottoinsieme di processori (specificato per esempio tramite condizioni sul contenuto della memoria locale) l'esecuzione di una determinata istruzione — la stessa per ogni processore — che può riguardare la manipolazione del contenuto dei registri, della memoria o il passaggio di dati ad altri processori. A questa categoria di elaboratori si possono ricondurre gli array sistolici, dove i processori sono disposti su una griglia bidimensionale, e la Connection Machine, che dispone i processori sui vertici di un ipercubo realizzando un meccanismo di *routing* che permette l'indirizzamento diretto di ogni processore da parte di ogni altro. In ogni caso, implementazioni di questo tipo dovrebbero prevedere la possibilità di operare in modo virtuale, cioè con un numero di nodi non limitato dal numero di processori realmente disponibili.

Esistono anche implementazioni VLSI di modelli connessionisti. Ricordiamo il chip sviluppato al Caltech contenente 289 nodi completamente connessi in grado di realizzare il modello di Hopfield, e il chip della AT&T realizzato con tecnologia mista analogico/digitale e in grado di funzionare anch'esso come memoria associativa. Una differente tecnologia che offre a media scadenza grandi possibilità in questo campo è sicuramente quella ottica [9]: in un computer ottico il portatore fondamentale di informazione non è l'elettrone ma il fotone. Questo comporta enormi vantaggi rispetto al passaggio di informazione fra

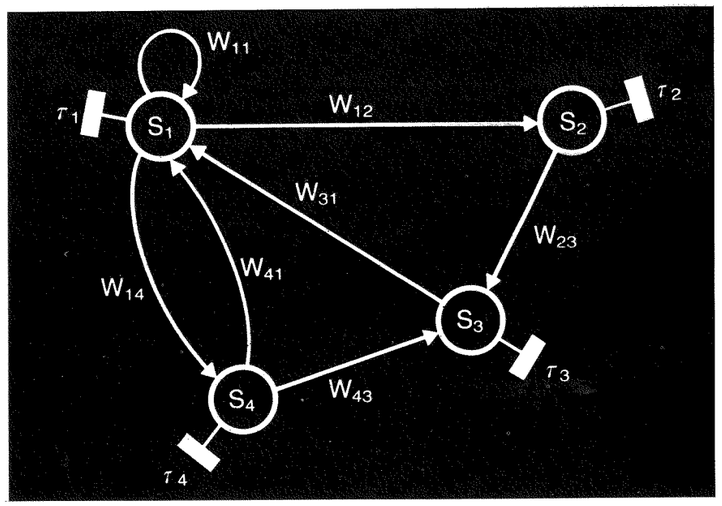


Fig. 4 - Un esempio di rete ciclica. Il suo grafo di rappresentazione contiene dei cammini chiusi (per esempio $S_1 \rightarrow S_4 \rightarrow S_3 \rightarrow S_1$).

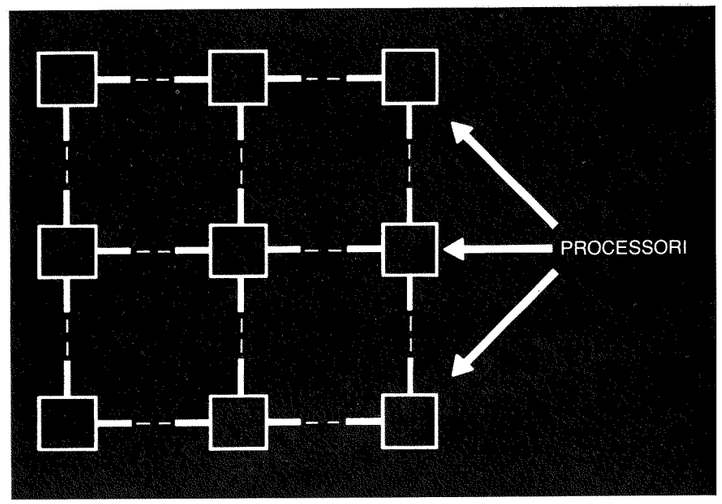


Fig. 5 - Un array cellulare bidimensionale. Ogni processore dispone di una piccola RAM e può inviare dati ai suoi vicini più prossimi.

elementi del sistema (assai importante per le reti neuronali) e la possibilità di eseguire in parallelo un numero altissimo di operazioni elementari. A più lunghe scadenze, sono da tenere anche in conto gli sviluppi degli elaboratori molecolari, un tipo di elaboratore "organico" nel quale i dati sono codificati nella struttura di proteine che vengono opportunamente modificate dall'azione di enzimi che rappresentano i processori del sistema.

7. Un'appendice neurologica

Le tecniche di manipolazione simbolica, proprio perché si basano su un insieme predefinito e poco flessibile di pochi simboli astratti, hanno rivelato una difficoltà congenita riguardo all'apprendimento e al pattern recognition riferiti ad un universo di forme naturali (cioè non artefatte come i solidi geometrici).

Una curiosa dimostrazione della difficoltà ad interagire con la realtà con un ragionamento unicamente basato sulla manipolazione di simboli predefiniti è descritta in un'opera del neurologo inglese Oliver Sacks [10]; essa riguarda il caso di un uomo che aveva subito una lesione alla corteccia cerebrale dell'emisfero destro.

Come è noto, la corteccia è la parte del cervello filogeneticamente più recente. Essa è suddivisa — come tutta la massa cerebrale — in due emisferi. I recenti progressi nel campo della neurologia e della più giovane neuropsicologia hanno permesso di individuare le aree della corteccia preposte ad alcune capacità cognitive. Senza addentrarci nell'argomento, possiamo semplificare enunciando le funzionalità dei due emisferi nel modo seguente: l'emisfero sinistro appare prevalentemente coinvolto nei processi di tipo razionale e analitico — il cosiddetto emisfero "scientifico", quello destro nei processi di tipo intuitivo e sintetico — il cosiddetto emisfero "artistico".

Il paziente in questione presentava una sindrome estremamente curiosa: l'uomo era affetto da *prosopagnosia,* cioè aveva perso la capacità di riconoscere i volti. Un danno nell'emisfero destro aveva compromesso il processo cognitivo che permette di riconoscere "a colpo d'occhio" un volto noto; potremmo dire che egli aveva perduto quella percezione diretta, immediata delle cose che ci circondano. Infatti, i sintomi coinvolgono anche il riconoscimento di oggetti, scene o immagini: l'uomo, per esempio, non era in grado di riconoscere una rosa, un paio di guanti o una fotografia.

A questo deficit cercava di venire in aiuto l'emisfero sinistro il quale, però, poteva contare *soltanto* sulle proprie capacità di manipolazione simbolica. Osservando il viso di un amico egli non lo vedeva nella sua totalità ma, piuttosto, come una collezione di tratti caratteristici: occhi, orecchie, naso, bocca. Se qualcuno di questi tratti era anomalo, per esempio il viso era deturpato da una cicatrice, allora egli riusciva, appoggiandosi sui propri ricordi simbolici, a riconoscere il suo amico "con quella grossa cicatrice". Ma, in ogni caso, il riconoscimento era sempre titubante e non accompagnato dalla sicurezza con la quale siamo in grado di svolgere il medesimo processo per i volti a noi più noti. I limiti dell'emisfero sinistro si rivelavano altrettanto drammaticamente anche nel caso di riconoscimento di oggetti. Di fronte ad una rosa e alla domanda di quale oggetto si trattasse l'uomo rispose:

"Quindici centimetri circa di lunghezza, una forma rossa convoluta con un'appendice lineare verde..."

Di fronte ad un paio di guanti osservò:

"Una superficie continua, avvolta su sè stessa. Dotata di cinque estensioni cave, se così si può dire..."

D'altra parte, egli non aveva alcuna difficoltà a riconoscere dei solidi geometrici, anche di forma complessa, proprio per la loro natura "platonica", cioè esplicitamente formale. Le capacità percettive esibite da questa persona sembrano indicare come, per arrivare ad una corretta comprensione della realtà, non sia sufficiente un processo di scomposizione analitica nel quale si cerca di assegnare un simbolo noto ad ogni elemento costitutivo di una "scena percettiva", sia essa visiva o ottenuta da altre fonti. Sacks osserva come il tipo di comportamento riscontrato in presenza di questo deficit cerebrale possa essere paragonato alle prestazioni degli algoritmi, soprattutto nel campo dell'Intelligenza Artificiale, che cercano di simulare le capacità percettive degli esseri umani.

I modelli connessionisti non pretendono certamente di sostituire le tecniche di ragionamento simbolico nel campo dell'emulazione delle capacità cognitive ma, piuttosto, di integrarsi strettamente con esse (e questo è il vero traguardo) per dare origine a elaboratori in grado di tener testa alla varietà e mutevolezza del reale attraverso un'interazione costruttiva con esso, o, per citare le parole di Teuvo Kohonen, "*...complying with the true statistical nature of the environment and its events is the most important task for these devices"* [11].

Bibliografia

[1] W.S. McCULLOCH e W.H. PITTS: A logical calculus of ideas immanent in nervous activity, *Bull. Math. Biophys.,* 2 (1943)115-133.

[2] D.O. HEBB: The Organization of Behaviour, Wiley, New York, 1949.

[3] F. ROSENBLATT: Principles of neurodynamics, Spartan, New York, 1962.

[4] M. MINSKY e S. PAPERT, Perceptrons, MIT Press, Cambridge MA, 1969.

[5] I.E. RUMELHART, G.E. HINTON and R.J. WILLIAMS: Learning internal representations by error propagation, *Institute for Cognitive Science technical report 8506,* UCSD, La Jolla, CA 92093.

[6] J.J. HOPFIELD: Neural networks and physical systems with emergent collective computational abilities, *proc. Natl. Acad. Sci USA,* 79 (1982)2554-2558.

[7] G.E. HINTON, T.J., SEJNOWSKI e D.H. ACKLEY: Boltzmann Machines, Constraint Satisfaction Networks that Learn, *Department of Computer Science technical report CMU-CS-84-119,* CMU, Pittsburgh, PA 15213.

[8] J.H.M. KORST e E.H.L. AARTS: Combinatorial Optimization on a Boltzmann Machine, *proc. of European Seminar on Neural Computing,* London, February 1988.

[9] Y.S. ABU-MOSTAFA e D. PSALTIS: Il calcolatore ottico neuronico, *Le Scienze,* 225 (1987)52-62.

[10] O. SACKS: L'uomo che scambiò sua moglie per un cappello, Adelphi, Milano, 1986.

[11] T. KOHONEN: Self-Organization and Associative Memory, Springer-Verlag, Berlin, 1988 (2nd edition).

Un Sistema Esperto per la riconfigurazione di mainframe

DONATO TOPPETA

Honeywell Bull Italia
Direzione Informatica Aziendale ed Organizzazione
Milano

1. INTRODUZIONE

L'articolo presenta un Sistema Esperto, denominato HOPE, sviluppato nell'ambito dei servizi di automazione interna della Honeywell Bull Italia.

HOPE è l'acronimo di "Honeywell Bull Operator Enhancer": sistema esperto per il supporto ad un operatore di mainframe HB DPS8 nelle procedure di riconfigurazione.

Obiettivo primario di HOPE è stata la verifica sperimentale dell'impatto tecnologico e strategico sul processo di definizione e sviluppo di soluzioni informatiche derivate dall'integrazione tra competenze proprie dell'area dell'ingegneria del software con tecniche, metodologie e strumenti tipici dell'Intelligenza Artificiale.

Numerosi relatori al recente congresso di Intelligenza Artificiale IJCAI87 (in particolare nel corso dei dibattiti tra produttori, venditori ed utenti di Sistemi Esperti e strumenti per il loro sviluppo) [1] hanno sottolineato la rilevanza strategica dell'integrazione tra Sistemi Esperti e sistemi informativi aziendali e la necessità di estendere le possibilità di utilizzo dell'Intelligenza Artificiale in ambito meno specialistico. Analoghe valutazioni si trovano anche in altre autorevoli fonti [2, 3, 10, 17].

HOPE è attualmente in uso effettivo presso i centri di Gestione Operativa della Honeywell Bull Italia situati a Borgolombardo e Caluso. Esso ha dimostrato concretamente che una persona di cultura universitaria, senza precedenti esperienze nell'area dell'I.A., può oggi realizzare su un hardware generico (PC in ambiente MS-DOS) un sistema esperto, in linguaggio PROLOG, di dimensioni, costo e tempo di sviluppo contenuti.

L'uso effettivo di HOPE ha consentito il ritorno economico dell'investimento effettuato e creato le premesse per una più estesa integrazione della tecnologia dei Sistemi Esperti nel MIS (Management Information System) aziendale.

Nel seguito dell'articolo vengono descritte concisamente la natura del problema, appartenente alla classe dei sistemi per la generazione di procedure di riconfigurazione dinamica, e le caratteristiche tecniche della soluzione.

L'ingegnere della conoscenza (la persona specializzata nello sviluppo di S.E.) ritroverà nel testo dell'articolo non solo alcuni paradigmi noti per la rappresentazione del dominio applicativo e per la ricerca delle soluzioni, ma anche alcuni motivi di interesse per soluzioni non classificabili nelle architetture più diffuse. Esse sono il risultato di uno sviluppo guidato dall'approfondimento delle caratteristiche del problema e dei processi decisionali dell'esperto umano.

Il dominio applicativo scelto per HOPE ha in letteratura numerosi riferimenti, sia in relazione all'uso di S.E. per il supporto alla Gestione Operativa di centri di elaborazione dati [4, 5, 6, 7], sia relativi alle problematiche di configurazione di elaboratori [5, 8].

Lo specifico problema si colloca tra quelli compresi nell'intersezione tra classi di soluzioni informatiche [2, 9, 16] che:

1) rispondono ad una reale esigenza dell'utenza,
2) sono tecnicamente fattibili con costi contenuti,
3) richiedono l'uso della tecnologia dei S.E. per la natura euristica del problema.

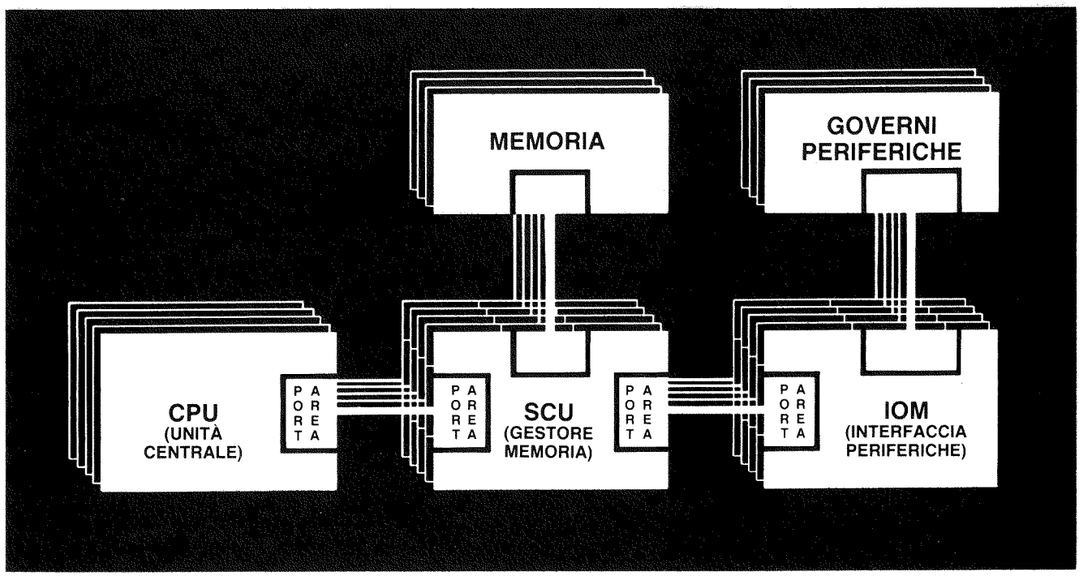
2. IL PROBLEMA

2.1 Descrizione sintetica

Gli elaboratori di grandi dimensioni (mainframe) DPS8 dell'Honeywell Bull hanno una struttura modulare multiprocessor che consente di realizzare differenti configurazioni dei componenti base [fig. 1]:

a) CPU (Central Processor Unit): contiene l'unità di calcolo aritmetico logico.
b) SCU (System Control Unit): controlla l'accesso concorrente delle CPU ed IOM ai banchi di memoria centrale.
c) IOM (Input/Output Multiplexer): interfaccia tutti i dispositivi di I/O (periferiche).

Un DPS8 è formato da almeno una CPU, una SCU ed un IOM, ma è possibile configurare su un singolo



[fig. 1]: La struttura di base di un mainframe DPS8 è costituita da componenti funzionalmente differenti, tra loro interconnessi:
a) CPU (Central Processor Unit): contiene l'unità di calcolo aritmetico/ logico.
b) SCU (System Control Unit): controlla l'accesso concorrente delle CPU ed IOM ai banchi di memoria centrale.
c) IOM (Input/Output Multiplexer): interfaccia tutti i dispositivi di I/O (periferiche).
Un DPS8 è formato da almeno una CPU, una SCU ed un IOM, ma è possibile configurare su un singolo DPS8 fino a 4 SCU e 8 tra CPU ed IOM (massimo 6 CPU).

DPS8 fino a 4 SCU e ad 8 tra CPU ed IOM (massimo 6 CPU). [fig. 2].

I collegamenti tra i componenti sono realizzati mediante connessioni fisiche [fig. 2] tra i "port" (interfacce) dei componenti stessi e simultaneamente sono descritti in forma simbolica congruente nei parametri di installazione dell'elaboratore (flusso di configurazione) letti dal S.O. (Sistema Operativo) GCOS8. Tali collegamenti (4 per ogni CPU e IOM, 8 per ogni SCU) sono inoltre soggetti a specifici vincoli.

I DPS8 formati da più di un CPU, SCU, IOM consentono di soddisfare esigenze di potenza elaborativa maggiore e possono funzionare in configurazione ridotta, nel caso di guasto di una o più componenti ridondanti, mediante "riconfigurazione".

Le operazioni di riconfigurazione richiedono sempre il cambiamento di diversi commutatori corrispondenti fra l'altro ai port dei componenti. Nel caso che sia possibile effettuare la riconfigurazione in forma dinamica, ossia senza "spegnere" il DPS8, è necessario eseguire procedure costituite da azioni hardware e da comandi software. Tali procedure in alcuni casi sono molto complesse, parametriche e critiche per il rischio di "caduta" del DPS8 (ossia l'improvvisa interruzione del servizio con conseguenti costi e tempi di ripristino della congruenza dei dati e di mancato servizio all'utenza).

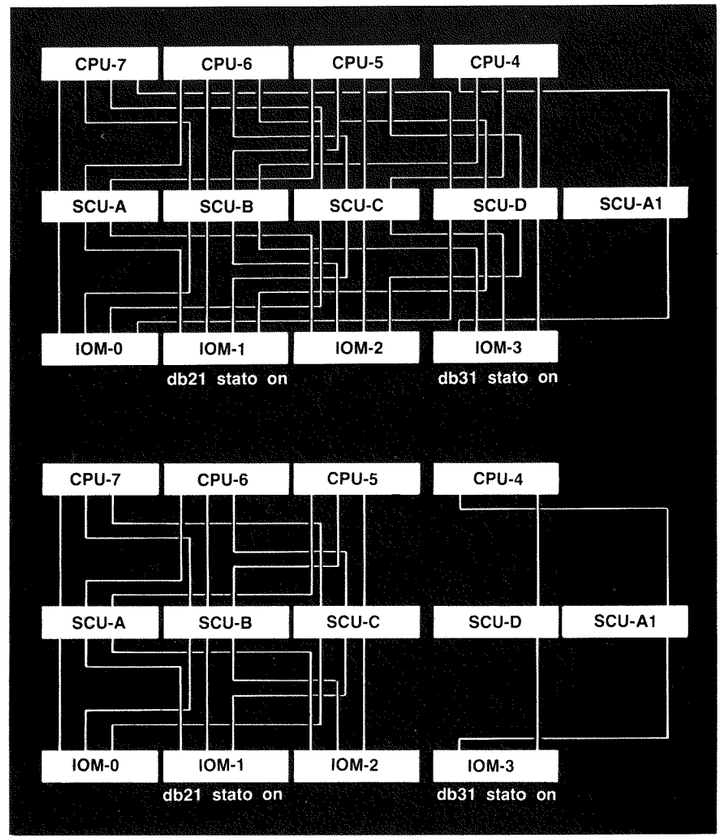
Le procedure di riconfigurazione dipendono anche dallo stato di effettiva configurazione fisica dell'elaboratore DPS8, il quale è uno dei possibili stati di configurazione, definiti implicitamente nel flusso di configurazione, e raggiungibili per riconfigurazione.

Il problema trattato è reso più complesso da alcuni aspetti specifici delle architetture di configurazione adottate: sono state infatti definite alcune interconnessioni tra DPS8 separati.

Ciò consente di trasferire rapidamente alcuni componenti (ma non tutti) da un sistema ad un altro, purchè interconnesso.

In determinate situazioni questa operazione può essere effettuata in

[fig. 2]: Esempio di configurazione reale di due sistemi DPS8 interconnessi; il primo, chiamato DB21, è normalmente costituito da 3 CPU, 3 SCU e 3 IOM, il secondo, chiamato DB31, è formato da 1 CPU, 2 SCU ed 1 IOM. Il grafo delle connessioni tra i port dei componenti rappresentato in alto si riferisce a tutte le connessioni definite nel file di configurazione. Solo alcune delle connessioni possono essere fisicamente abilitate simultaneamente; nel grafo inferiore sono riportate solo quelle normalmente attive. È possibile cambiare lo stato delle connessioni (e dei componenti) mediante riconfigurazione, ottenendo differenti stati di configurazione dei due sistemi. Entrambi i grafi sono stati generati e stampati da HOPE; gli originali sono a colori per agevolarne la comprensione.



forma dinamica. In questo modo è possibile adeguare la ripartizione della capacità di calcolo dei diversi DPS8 alle esigenze gestionali, in funzione del carico dei sistemi, della loro criticità e dell'affidabilità dei componenti. Questa scelta comporta un consistente aumento della complessità del sistema risultante e richiede competenze particolari.

2.2 Obiettivo

L'obiettivo del S.E. HOPE è quello di supportare operatori di media esperienza nelle operazioni di riconfigurazione di un mainframe DPS8 Honeywell Bull limitatamente alle operazioni di rilascio, inserimento e trasferimento dei suoi componenti centrali (CPU, SCU, IOM). Ciò comporta per l'azienda una riduzione dei tempi di interruzione dei Servizi Informativi centralizzati ed una migliore qualità del servizio e dell'ambiente di lavoro. Collateralmente si è riorganizzata e formalizzata la conoscenza dell'esperto di riconfigurazione agevolandone l'aggiornamento, la consultazione ed il confronto con altre fonti, anche a scopo didattico.

L'obiettivo primario è stato così scomposto:

1) Individuare per qualsiasi configurazione esistente, tra quelle possibili, la sequenza dettagliata e corretta di azioni hardware e software per eliminare, inserire o trasferire un componente.
2) Garantire la correttezza della configurazione risultante rispetto ai vincoli di configurazione eliminando i componenti incompatibili con la nuova configurazione.
3) Individuare per ogni operazione lo stato attuale e futuro dell'elaboratore.
4) Adeguare la rappresentazione interna della configurazione reale in funzione delle operazioni richieste.
5) Fornire una rappresentazione esaustiva, sintetica, dinamica e familiare della configurazione attuale dell'elaborazione in forma grafica, a video e su stampante, e di report.
6) Garantire la manutenibilità del sistema verso nuove configurazioni o revisioni delle procedure operative, mediante la definizione di uno specifico linguaggio di descrizione della conoscenza comprensibile e modificabile dall'esperto del dominio applicativo.
7) Creare i presupposti per un più agevole ed oggettivo confronto e scambio di esperienza tra esperti di riconfigurazione diversi mediante la rappresentazione delle loro conoscenze realizzata da HOPE.

2.3 Criticità del problema

Il valore del S.E. HOPE è legato intrinsecamente alla elevata criticità delle operazioni per cui viene fornito supporto agli operatori.

A) Durante il giorno i mainframe dei Servizi Informativi Interni dell'Honeywell Bull sono dedicati prevalentemente ad applicazioni TP (sistemi gestionali in tempo reale) e di natura diversa (posta elettronicamente, Inuiry, tools end-user, Time Sharing, sviluppo applicativo, etc.) con un numero elevato di utenti simultanei (fino a 100) connessi via network.

   In tale situazione è fondamentale:

1) Ridurre i rischi di blocco del S.O. del mainfreame (perchè il costo del ripristino di correttezza dei dati ed il numero di utenti coinvolti nell'interruzione del servizio sono elevati).

2) Accelerare i tempi di esecuzione delle procedure (nelle ore diurne l'esperto è normalmente reperibile ma può non esserlo immediatamente od aver bisogno di tempo per verificare lo stato di configurazione o la sintassi dei comandi, avendo molti altri compiti ed eseguendo molto raramente alcune specifiche procedure).

3) Operare in modo pianificato pur in situazioni impreviste, caratterizzate da elevato stress ambientale.

B) Durante la notte gli stessi elaboratori eseguono prevalentemente elaborazioni a lotti, senza utenti connessi in tempo reale. In questa situazione la criticità delle operazioni di riconfigurazione è determinata dai problemi di reperibilità dell'esperto presso il suo domicilio. L'esperto ha sempre visto con favore uno strumento che riduca la necessità di un suo intervento diretto di notte, motivandolo ulteriormente ad una attiva partecipazione allo sviluppo del sistema Esperto.

Un ulteriore fattore di criticità della conoscenza formalizzata da HOPE è di tipo organizzativo: le caratteristiche globali del problema e la specificità di ogni centro di calcolo rendono complesso e lento l'accumulo e lo scambio di esperienza reale e quindi critica la possibilità di disporre di operatori di consolle esperti di riconfigurazione in ogni turno di lavoro.

Infatti, riconfigurare il sistema è un'operazione che può richiedere da pochi minuti fino a circa mezz'ora; ciò rende il costo di una esercitazione troppo elevato dovendo, in tale contesto, dedicare un mainframe alla funzione di addestramento.

2.4 Dimensioni e oggetti del problema

Scopo di questo paragrafo è dare un'idea del grado di complessità del problema e del numero di "oggetti" trattati mediante alcuni parametri quantitativi e qualitativi.

Il sistema HOPE è stato inizialmente sviluppato per il centro di elaborazione dati Honeywell Bull Italia di Borgolombardo dove sono installati 4 DPS8 per un totale di 7 CPU, 9 SCU, 7 IOM con 144 port di cui 112 predisposti per le connessioni tra componenti. È stato in seguito esteso ed adeguato al Centro elaborazione dati HBI di Caluso.

Le azioni hardware effettuate nei processi di riconfigurazione sono genericamente:

- abilitare/disabilitare port sui pannelli dei componenti (fisicamente separati);

- regolare rotary switch ed altri commutatori.

Le azioni software sono costituite da messaggi parametrici da introdurre su una specifica consolle di sistema riferiti a comandi di rilascio/inserimento "logico" di componenti ed a funzioni e test ad essi accessori (ad esempio sospensione del programma di monitoraggio degli errori ECOL).

3. SOLUZIONE

3.1 Architettura interna

Il problema affrontato da HOPE si presenta esternamente come procedurale, ma il lettore avrà individuato nella descrizione precedente che ogni azione di riconfigurazione è il risultato di un processo mentale complesso, dipendente da una comprensione profonda dell'architettura interna dell'elaboratore DPS8, del suo stato e dei suoi vincoli.

L'idea centrale di HOPE è stata pertanto di individuare una rappresentazione di tutti i parametri di configurazione ed utilizzare tale informazione generando per inferenza le procedure di riconfigurazione.

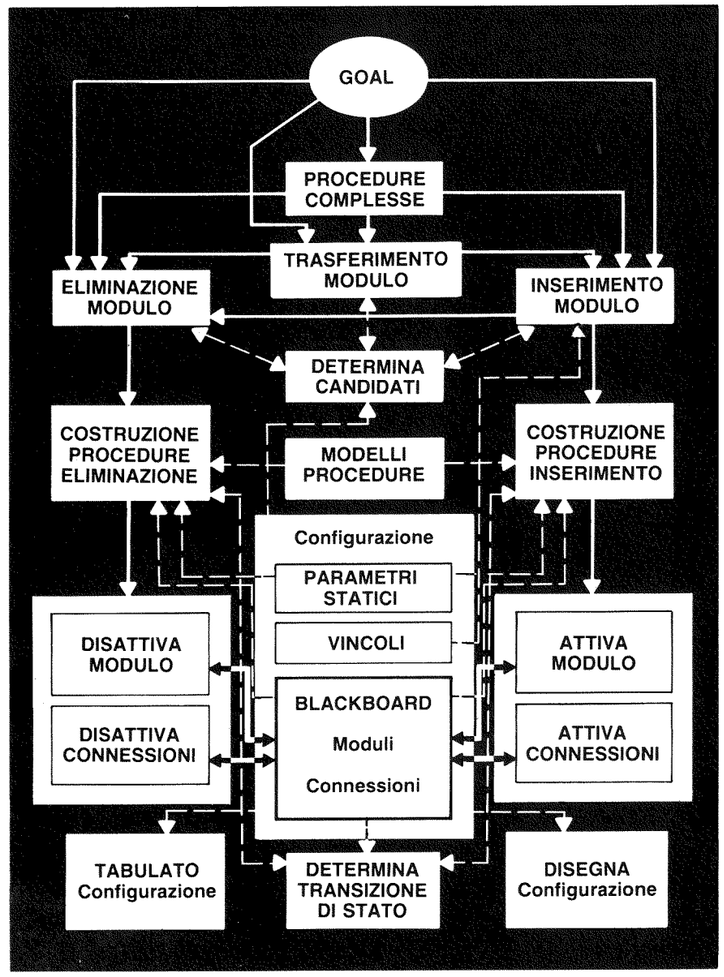
La tecnologia dei Sistemi Esperti ha consentito di descrivere in forma dichiarativa i parametri e i vincoli di configurazione sfruttando le capacità di base dati relazionale integrata nel linguaggio PROLOG [11, 12]. Sono state definite delle tabelle relazionali in cui sono elencati i componenti delle specifiche configurazioni dei DPS8, le loro connessioni, i vincoli ed alcune informazioni accessorie.

Questa struttura contiene l'informazione dei file di configurazione e dei grafi di connessione dei componenti del DPS8; essa può essere direttamente compresa, verificata ed aggiornata dall'esperto del dominio applicativo.

Inoltre i records delle tabelle relazionali riflettono, nei campi variabili, lo stato del sistema DPS8 riferito a tutti i suoi componenti non periferici (CPU, SCU, IOM e relativi port). È perciò possibile utilizzare la componente dinamica di tale rappresentazione per mantenere traccia delle operazioni di riconfigurazione effettuate [fig. 3].

Tutte le componenti inferenziali del S.E. accedono a tale struttura per estrarne informazioni (alcune anche per modificarla) realizzando una modalità di cooperazione e comunicazione tra meccanismi inferenziali diversi riconducibile per alcuni aspetti al modello di S.E. a lavagna ("blackboard") [18] [fig. 4].

[fig. 3]: In questo disegno è rappresentata in forma schematica e semplificata la struttura semantica iniziale di HOPE. È stato messo in particolare rilievo il ruolo centrale della Blackboard (lavagna) rispetto ai differenti meccanismi inferenziali utilizzati nel Sistema Esperto. È anche facilmente individuabile il meccanismo di scomposizione del goal (obiettivo) formulato dall'operatore nelle azioni necessarie a costruire la procedura di riconfigurazione.



L'architettura di S.E. a lavagna si riferisce alla classe di sistemi che utilizzano simultaneamente più meccanismi inferenziali e basi di conoscenza separati per gestire la consultazione dell'utente. Tale cooperazione viene resa trasparente all'utente e gestita dal sistema mediante la definizione di un'area comune di comunicazione tra i processi inferenziali. Questa soluzione rispecchia la modalità con cui un gruppo di esperti in aree diverse coopera alla soluzione di un problema mediante una lavagna su cui ognuno può leggere, scrivere o modificare quanto è già presente. Questo processo può essere riferito anche al metodo di ragionamento di un solo esperto che utilizza differenti fonti e modelli di conoscenza per risolvere un problema complesso.

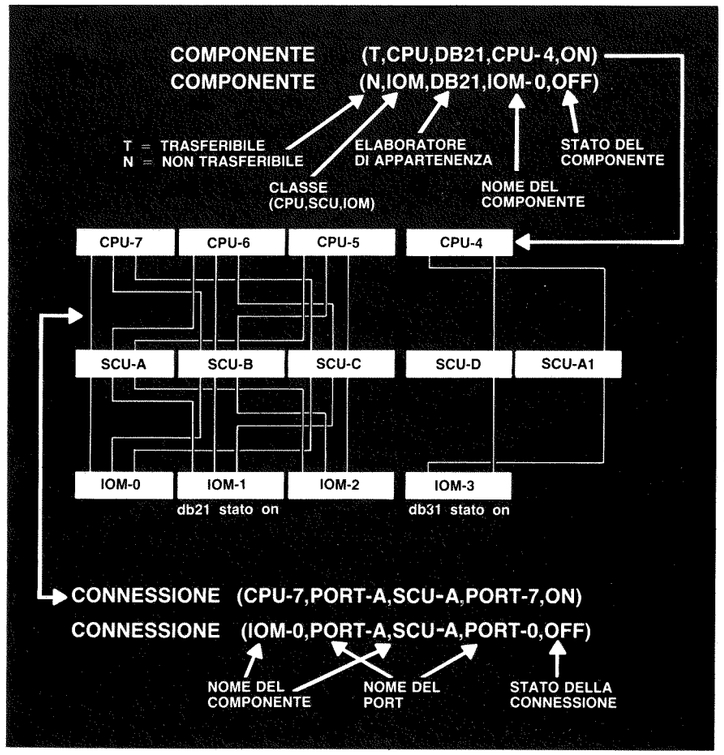
Un esempio di tali motori inferenziali, singolarmente molto semplici e come tali realizzati da pochi predicati PROLOG, è quello che presiede all'individuazione dell'azione di riconfigurazione richiesta dall'utente: il trasferimento di un componente da un DPS8 al suo gemello è scomposto nell'operazione di rilascio dal sistema a cui esso è connesso, seguita da quella di inserimento nell'altro DPS8.

In una particolare coppia di DPS8, prima di soddisfare la richiesta di inserimento di un nuovo componente, è necessario verificare se esistono componenti incompatibili (perchè privi di opportune connessioni a livello di port) ed in caso di risposta positiva attivarne la procedura di rilascio.

La selezione del componente su cui si vuole intervenire innesca invece una ricerca in blackboard di tutti i componenti che possono essere candidati all'operazione scelta (rilascio, inserimento, trasferimento) nello stato attuale di configurazione; viene così generato un menù su cui l'operatore effettua la sua scelta.

Tale ricerca prevede vincoli ulteriori nel caso di componenti definiti non trasferibili (solitamente IOM).

L'individuazione del potenziale stato presente e futuro (cioè dopo una operazione di riconfigurazione) del



[fig. 4]: La descrizione tabellare semplificata di alcuni componenti e connessioni dello stato di configurazione è messa in diretta relazione con la rappresentazione grafica della configurazione dell'elaboratore. Le due rappresentazioni sono perciò equivalenti: quella grafica è più sintetica e intuitiva per l'uomo, quella tabellare è più esatta e adatta ad essere trattata da HOPE, ma è comunque ancora perfettamente riconoscibile perchè codificata in forma dichiarativa.

DPS8 interessato avviene mediante un apposito meccanismo inferenziale associato a sei predicati PROLOG (nella nuova versione l'utente può informare il sistema di aver spento un DPS8 potenzialmente attivo).
La ricerca dello stato è la trascrizione della definizione di "DPS8 potenzialmente attivo" se "esistono almeno un CPU, SCU, IOM attivi ed appartenenti al sistema" mediante ricerca in blackboard. La ricerca dello stato di un DPS8, ad esempio successiva all'inserimento di un componente, prevede che un sistema incompleto possa essere attivato se viene inserito un componente della sola classe mancante.
L'individuazione dello stato attuale e di quello successivo a riconfigurazione determinano ad esempio la necessità di start-up (partenza) e shut-down (spegnimento) di un DPS8, e l'esigenza di introdurre comandi rivolti al S.O. (inutili ed impossibili quando l'elaboratore è spento).

Le informazioni fornite dall'utente sul tipo di operazione richiesta e sul componente interessato vengono completate con quelle estratte direttamente o dedotte dalle informazioni di stato di configurazione (DPS8 interessato, stato attuale e futuro, etc.), creando una chiave di selezione della "procedura scheletro" da eseguire.

Il termine "procedura scheletro" si riferisce ad un modello parametrico di tipo procedurale di un sottoinsieme delle procedure di riconfigurazione che, completato con i valori variabili (chiavi di selezione o valori ottenuti per inferenza sulla blackboard utilizzando tali parametri), costituisce la specifica istanza di procedura di riconfigurazione.

La selezione della procedura di riconfigurazione mediante chiave di selezione utilizza le capacità di pattern-matching del linguaggio PROLOG. Il pattern-matching (letteralmente "confronto di forme") avviene tra i valori delle variabili istanziate della chiave di selezione ed i valori della corrispondente lista che caratterizza il dominio di validità della procedura scheletro.

Nel caso specifico è possibile definire il valore di quantificatore universale (ossia di jolly) per uno o più campi della lista che caratterizzano le procedure scheletro (che diventano così riferite a casi più generali).

Ciò consente di realizzare differenti livelli di aggregazione delle singole istanze di procedura, da classi molto generiche (ad esempio rilascio di un qualsiasi componente con DPS8 spento) a singoli casi relativi ad eccezioni.

In questo modo è possibile astrarre le procedure scheletro dagli elementi specifici della configurazione, focalizzando in esse la componente di decisione procedurale dell'esperto. Ad esempio l'operatore deve usare un diverso comando di consolle per il rilascio di una CPU rispetto ad un SCU o IOM. Tale comando viene poi completato con parametri che

l'esperto decide in funzione dello stato di configurazione (indirizzo iniziale di memoria, etc.).

Il modello inizialmente adottato per descrivere le procedure scheletro ed attualmente in uso effettivo prevedeva una struttura articolata in "azioni e test iniziali", "azioni sui port", e "azioni e test conclusivi".

La prima e l'ultima fase erano costituite da una sequenza rigida di azioni e comandi presentati all'operatore come messaggi seguiti da richieste di conferma.

Le azioni sui port sono invece il risultato di una ricerca di tipo relazionale sui port del componente interessato e su quelli ad esso connessi, coinvolti nell'operazione richiesta.

Ad esempio:

"Seleziona tutte le connessioni tra i port non attivi del componente da inserire ed i port di altri componenti attivi ed appartenenti all'elaboratore su cui verrà inserito il computer selezionato e rendile attive".

La rappresentazione interna (blackboard) viene in questo caso aggiornata in modo corrispondente all'intervento fisico sui port dei componenti richiesto dall'operatore. Questa è l'operazione più difficile per l'uomo il quale deve ricordarsi stato e connessioni di tutti i componenti del (o dei) DPS8 in modo esatto, selezionando quelli interessati. Questa problematica è comune alla gestione di strutture complesse connesse con una modalità a rete.

In seguito è stata definita una architettura più sofisticata per descrivere le procedure scheletro. Essa consente di gestire sequenze più complesse di risposte SI/NO alle domande proposte all'operatore da HOPE per verificare che il DPS8 reagisca correttamente all'operazione di riconfigurazione e che essa sia compatibile con i vincoli organizzativi (ad esempio effettuare alcune operazioni critiche solo se strettamente necessarie o se non vi sono utenti collegati).

Il modello di procedure scheletro sviluppato ed attualmente in fase di test si basa sul concetto di automa deterministico a stati finiti.

Ogni stato corrisponde ad una lista di messaggi costituiti da stringhe fisse o da variabili (nome del sistema DPS8, del componente o informazioni ad esso correlate) che vengono visualizzate sullo shermo.

In funzione della risposta dell'operatore è definito lo stato successivo; od, in alternativa, uno stato terminatore corrispondente al completamento della procedura di riconfigurazione od alla richiesta di contattare l'esperto per gestire una situazione che esula dalla conoscenza di HOPE (e qualche volta dell'esperto umano che risolve il problema con una strategia diversa).

Uno stato particolare è associato alle operazioni sulla blackboard precedentemente descritte come "azioni sui port".

La lunghezza e la sequenza delle azioni richieste all'operatore non è fissata a priori ma è soggetta ai vincoli dei percorsi, eventualmente ciclici, ammessi dagli automi in funzione delle risposte dell'operatore alle domande poste da HOPE.

Ogni procedura scheletro è costituita da uno o più automi riconoscitori (alcuni possono essere comuni a più procedure).

Lo stato iniziale è individuato per pattern matching tra la chiave di selezione della procedura scheletro, assegnata in fase di esecuzione, ed i parametri statici definiti (anche parzialmente) per individuare la classe di procedure di riconfigurazione a cui il modello si riferisce.

La componente procedurale della conoscenza di HOPE è descritta con questo linguaggio "astratto" in un flusso esterno che viene caricato in memoria utilizzando per la sua rappresentazione una struttura mista di:

1) tabelle relazionali (per l'individuazione dello stato iniziale per pattern matching sequenziale con la chiave di selezione);
2) albero binario (ordinato sul campo indirizzo simbolico che identifica univocamente ogni stato consentendo una ricerca molto efficiente);
3) liste (per non vincolare la lunghezza ed il numero dei messaggi associati ad ogni stato).

Il linguaggio PROLOG ha consentito di manipolare molto facilmente tali strutture, utilizzandone la capacità di binding dinamico (ossia la coerenza tra tipi di dati della procedura chiamante e di quella chiamata è verificata in fase di esecuzione e non di compilazione) e la natura ricorsiva (una procedura può richiamare se stessa allocando memoria in forma dinamica) per realizzare con pochi predicati un interprete del linguaggio di descrizione degli automi.

Questa soluzione è stata adottata perchè, oltre ad essere molto potente ed estendibile, consente all'esperto del dominio applicativo di comprendere e verificare anche la base di conoscenza procedurale di HOPE astraendo dalla maggior parte dei parametri di configurazione e mantenendo un totale controllo del dialogo con l'operatore.

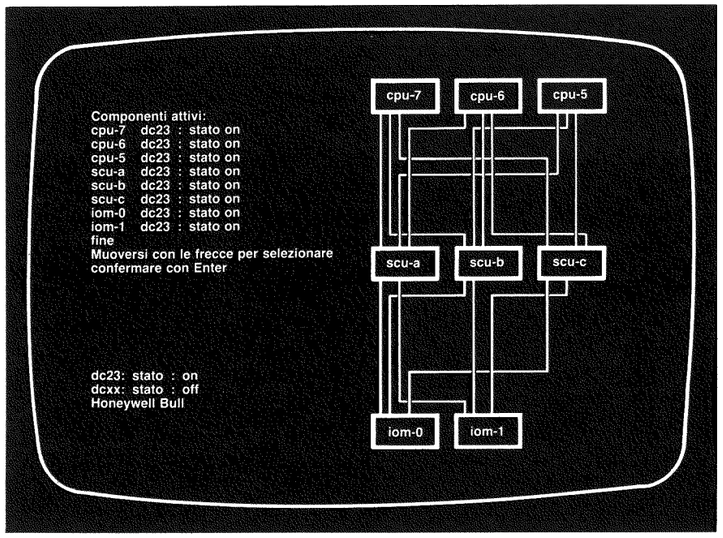
Infatti la conoscenza del processo di riconfigurazione è soggetta ad aggiornamenti causati da modifiche hardware (cambiamenti ai componenti ed alla configurazione adottata), software (nuove versioni del Sistema Operativo) ed organizzative (diverse ripartizioni del carico di lavoro sui diversi sistemi).

Ciò rende praticamente impossibile mantenere una documentazione cartacea esauriente ed aggiornata delle specifiche procedure di riconfigurazione perchè il costo (ed il tempo) della loro stesura (e validazione) sarebbe troppo elevato rispetto alla frequenza di utilizzo delle singole procedure.

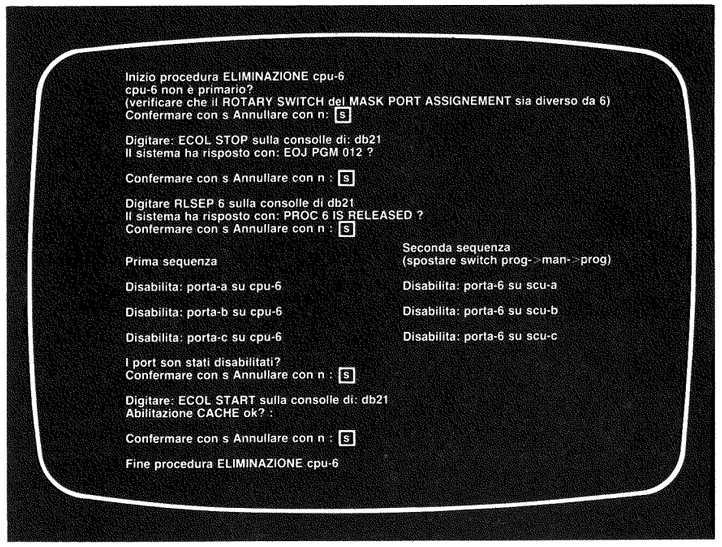
3.2 Interfaccia utente

L'interfaccia utente di HOPE è stata disegnata in stretta relazione all'ambiente di utilizzo.

La prima esigenza era quella di evitare ambiguità nelle risposte utente; la soluzione adottata si avvale di menu per la scelta dell'obiettivo primario della consultazione e del componente (e sistema) interessato al processo di riconfigurazione. Vengono visualizzate automaticamente solo le scelte ammissibili, l'operatore effettua la selezione con i tasti alto/basso e la conferma con ENTER [fig. 5]. I test delle procedure sono strutturati con risposte SI/NO [fig. 6].



[fig. 5]: Esempio di una videata di HOPE relativa ad un sistema DPS8 differente dai precedenti. Sulla sinistra, HOPE ha generato, in risposta alla richiesta di rilasciare un componente del mainframe, l'elenco, strutturato a menù, dei componenti attivi che perciò possono essere rilasciati. Sulla destra, il grafo aggiornato dello stato del sistema agevola il processo di scelta.



[fig. 6]: Esempio di procedura di rilascio del CPU6 conseguente alla scelta effettuata sul menù di fig. 5. Questa è una delle più banali procedure che HOPE può generare. È comunque riconoscibile la presenza di una domanda per verificare se il componente da rilasciare richieda un trattamento particolare, la richiesta di comandi da effettuare in consolle e di azioni fisiche sugli "switch" dei "port".

La semplicità di questo approccio ha consentito di evitare problemi di comprensione, di contenere il costo di sviluppo e di avere tempi di risposta trascurabili (1-2 secondi massimo).

La seconda esigenza riguarda la comprensione dello stato di configurazione dei DPS8 prima e dopo una operazione di riconfigurazione. La soluzione adottata si è rivelata determinante nel conseguire l'immediata accettazione di HOPE.

HOPE gestisce a tale scopo lo schermo a colori del PC costruendo il grafo aggiornato delle connessioni di configurazione del mainframe mediante l'indirizzamento diretto per riga/colonna e l'insieme di caratteri grafici. Inoltre, il colore delle connessioni e dei componenti è associato dinamicamente con codifiche convenzionali (il range è di 16 possibili colori).

Questa interfaccia sostituisce il grafo disegnato a mano fino ad ora utilizzato, che ovviamente si riferisce solo allo stato di configurazione normale (ed ai potenziali collegamenti alternativi) e non allo stato reale di un sistema riconfigurato.

L'operatore ora è in grado di comprendere meglio lo stato del sistema e la sua evoluzione guardando il grafo piuttosto che consultando spiegazioni testuali.

La versatilità della specifica versione di PROLOG è ulteriormente confermata da una funzione sviluppata per HOPE che legge l'immagine del grafo generata per via inferenziale sul video e la traduce nella sequenza di caratteri di controllo necessari ad ottenere una stampa a colori.

3.3 Funzioni ausiliarie

L'utilizzabilità di un sistema esperto in un ambiente reale dipende spesso da aspetti funzionali non strettamente legati al concetto di I.A. ma di rilevante impatto pratico.

In HOPE ad esempio è fondamentale garantire l'allineamento tra stato di configurazione e rappresentazione interna. Ciò avviene a condizione che tutte le riconfigurazioni dei DPS8 siano guidate da HOPE (o comunque su di esso riportate).

Ne deriva il problema di salvare lo stato in caso di necessità di utilizzo del PC per altre funzioni o di interruzione dell'alimentazione. Una specifica funzione di PROLOG registra su un flusso di servizio lo stato di memoria del sistema e può ripristinarlo da qualsiasi stato in pochi secondi. È perciò possibile utilizzare HOPE per simulare una procedura di riconfigurazione prima di effettuarla sul mainframe.

3.4 Strumenti di sviluppo

La scelta dell'hardware di utilizzo effettivo è stata effettuata secondo i seguenti criteri:

1) Utilizzare un hardware convenzionale di basso costo e di facile utilizzo per l'operatore.
2) Garantire l'indipendenza del S.E. dal mainframe che potrebbe trovarsi in stato "critico".
3) Consentire lo sviluppo di un'interfaccia utente gradevole ed efficace.

Un personal computer di costo contenuto (HB XP con video a colori e disco rigido) si è dimostrato completamente adeguato sul piano funzionale.

In fase di sviluppo si è adottato un PC di maggior potenza (HB AP) per compensare la minore efficienza dell'ambiente interpretato; questo ha evitato problemi di conversione verso l'ambiente di utilizzo (limitati alla sola fase di compilazione) ed ha orientato le scelte di progetto a valorizzare le risorse specifiche del PC (ad esempio l'interfaccia grafica a colori).

Il linguaggio adottato, l'Arity Prolog, compatibile con lo standard di Edimburgo ma dotato di utili estensioni, si è confermato adeguato alle esigenze di sviluppo anche nella generazione di una versione compilata affidabile ed efficiente [13, 14, 15].

3.5 Approccio metodologico

La natura euristica della conoscenza, la mancanza di documenti formali relativi al processo di riconfigurazione nel suo complesso e le proprietà del linguaggio PROLOG (ambiente interpretato, approccio dichiarativo, sviluppo incrementale) hanno determinato e favorito la metodologia di sviluppo mediante prototyping, tipica dei progetti di Sistemi Esperti.

La pianificazione è perciò costituita dallo sviluppo, secondo scadenze fissate, di prototipi successivi. Essi, verificati con l'esperto di riconfigurazione, hanno consentito di confermare, ad ogni fase dello sviluppo, la fattibilità del progetto, di mantenere una buona aderenza alle esigenze reali, di concordare priorità e limiti di applicazione di HOPE. Ma soprattutto il confronto tra il prototipo e l'esperto si è rivelato determinante nel fare emergere e rendere esplicite conoscenze e processi mentali che erano rimasti inespressi nelle precedenti discussioni.

La particolare accentuazione che l'Intelligenza Artificiale pone sul processo di esplicitazione e rappresentazione della conoscenza condotto dall'ingegnere della conoscenza è, a parere dello scrivente, un contributo di grandi prospettive alla ricerca di soluzioni più efficaci al problema dell'ingegnerizzazione del software.

4. CONCLUSIONE

HOPE ha dimostrato che la tecnologia dei Sistemi Esperti può risolvere un problema limitato e specifico, ma concettualmente complesso, in modo più efficace e competitivo di un approccio tradizionale.

Oltre a realizzare un efficace strumento operativo, questo progetto costituisce motivo di interesse più ampio nell'ambito in cui è stato sviluppato.

Infatti, come esempio riuscito dell'impiego dei concetti di Intelligenza Artificiale, esso vale come stimolo alla ulteriore applicazione di queste nuove e potenti tecnologie di progetto.

RINGRAZIAMENTI

Allo sviluppo di HOPE hanno dato un contributo determinante ed entusiastico parecchie persone dei servizi di automazione interna della Honeywell Bull Italia.
A tutti va il riconoscimento e i ringraziamenti dello scrivente.

Bibliografia

[1] AA VV, *Proceedings of the Tenth International Joint Conference on Artificial Intelligence,* Morgan Kaufmann Publisher Inc., 1987.

[2] SILVERMAN B.G., *Expert Systems for Business,* Addison Wesley, 1987.

[3] WINSTON P.H., PREUDERGAST K.A., *The A.I. Business,* MIT Press, 1986.

[4] MILLIKEN K.R. et Al., *YES/MVS and the Automation of Operations for Large Computer Complexes*, IBM Systems Journal, Vol. 25, N. 6, 1986.

[5] AA VV, *Honeywell Bull: International Artificial Intelligence Symposium, Proceedings,* Novembre 1987.

[6] BRADEN B., *Bullish on AI?* Computer/Electronic Service News, June 1987.

[7] FURLAN S.E., *Se in banca entra l'IA* Linea EDP n. 45, 1986.

[8] ROLSTON D., *Un sistema esperto "configura" i grandi elaboratori,* Quaderni di Informatica 26 - 1986.

[9] WINSTON P.H., *Artificial Intelligence,* II ed, 1984.

[10] AA VV, *Atti Congresso AICA: Intelligenza Artificiale e Ingegneria del Software: situazione e prospettive,* AICA-AEI, 1987.

[11] BRATKO I.: *Prolog Programming for Artificial Intelligence,* Addison Wesley 1986.

[12] MARCUS C., *Prolog Programming Application for database Systems, Expert Systems and Natural Language Systems,* A.W., 1986.

[13] *The Arity/Prolog Programming Language - Building Arity/Prolog Applications,* Arity Corporation, 1986.

[14] TORQUATI F., *Un sistema esperto per la diagnosi di elaboratori,* Quaderni di Informatica, 30, 1987.

[15] COVINGTON M., VELLINO A., *Prolog Arrives,* Tech Journal, November, 1986.

[16] WATERMAN D.A., *A Guide to Expert systems, Addison Wesley, 1986.*

[17] HARMON P., *Market analysis,* Expert Systems strategies Vol. 3 N. 6, 1987.

[18] NEWTON S.L., *Programming with P-Shell,* IEEE Expert, 1986.

Criteri di ricerca in grandi basi di dati multimodali

LEONARDO FELICIAN,
DARIO FACCHIN

Dipartimento di Matematica e Informatica
Università di Udine

1. INTRODUZIONE

Il rapido sviluppo delle tecnologie informatiche ha portato ad un continuo abbassamento dei costi e ad un continuo aumento di prestazioni dei dispositivi hardware.

Si sono aperti così nuovi campi applicativi, che richiedono lo studio e lo sviluppo di nuove soluzioni: il vero freno ad una più rapida espansione dell'uso di strumenti informatici deriva infatti, già da tempo, dalla relativa lentezza con cui vengono sviluppate le nuove applicazioni. Se agli albori dell'era informatica l'attenzione era incentrata sulla macchina, ben presto si è capito che il lavoro più pesante consiste nel progettare sistemi validi ed affidabili che forniscano un'interfaccia di livello sempre più alto alla macchina; come rilevato da tutti gli studi, i costi del software aumentano in maniera molto più rapida di quelli dell'hardware.

In questo lavoro si intende descrivere una soluzione per il reperimento dei dati in un sistema di archiviazione multimodale (*multimedia data base*), le cui linee architetturali sono state già presentate in un precedente articolo [1]. Un multimedia data base è una base di dati che contiene un insieme di oggetti, denominati documenti multimodali o semplicemente documenti, costituiti da diversi tipi di informazioni:

- dati strutturati
- testo
- immagini
- voce

Un esempio di multimedia data base, preso in considerazione nel seguito, è costituito dal sistema di archiviazione di documenti d'ufficio, che si compongono di una parte strutturata (nome, data di creazione, autore, provenienza, destinazione, etc.), di una parte testuale, intercalata da immagini (figure, grafici, fotografie, etc) e di una parte vocale (commenti a voce, registrazioni telefoniche, etc). Alcune di queste parti possono mancare.

Il motivo per cui si considera separatamente la parte testuale da quella immagine (pur essendo entrambe non strutturate) deriva dal fatto che memorizzare i testi come immagini bit-mapped risulta antieconomico: il costo di memorizzazione di una pagina in formato immagine è infatti circa dieci volte superiore rispetto ad una pagina di testo.

Un multimedia data base può contenere un numero molto elevato di documenti, per cui uno dei problemi più importanti è la ricerca efficiente dei dati. In generale, in ogni struttura di base di dati bisogna considerare da un lato l'organizzazione architetturale, cioè il disegno, la costruzione e gli algoritmi che operano il mantenimento del data base; dall'altro gli algoritmi per il suo utilizzo, in particolare per la funzione di reperimento dei dati (*retrieval*). Una migliore architettura implica maggiori investimenti, ma si traduce in un più efficiente utilizzo.

Nei tipici ambienti applicativi per le basi di dati questo principio ha portato a configurare organizzazioni in cui i dati sono altamente strutturati. Un esempio è costituito dalle basi di dati aziendali, nelle quali l'organizzazione dei dati, descritta da schema e sottoschemi, modella il sistema informativo dell'azienda stessa. Se tutto ciò porta innegabili vantaggi in termini di efficienza delle transazioni, non si possono disconoscere anche alcuni svantaggi.

Un'architettura molto strutturata risulta anche molto rigida e mal adatta ai cambiamenti imposti dalla continua evoluzione delle necessità applicative. Ciò è dimostrato dalla metodologia di sviluppo dei sistemi informatici in azienda: dapprima si svolge un'analisi al fine di costruire un modello (una fotografia dell'azienda ad un dato instante), poi, in secondo momento, si passa ad automatizzare le varie funzioni sulla base del modello. Il limite di questo approccio è costituito dal fatto che il sistema informativo non può essere rigido, ma deve evolvere continuamente, mentre il sistema informatico costruito a partire dal modello tende a congelare un ambiente che è destinato ad allontanarsi sempre più dalle varie esigenze, finchè pesanti e costose ristrutturazioni si rendono necessarie ed inevitabili, poichè in questi ambienti l'efficienza operativa costituisce un requisito irrinunciabile.

Un altro svantaggio delle architetture molto strutturate va ricercato nella difficoltà di accesso ai dati da parte dei non addetti ai lavori: l'utente deve conoscere a fondo l'organizza-

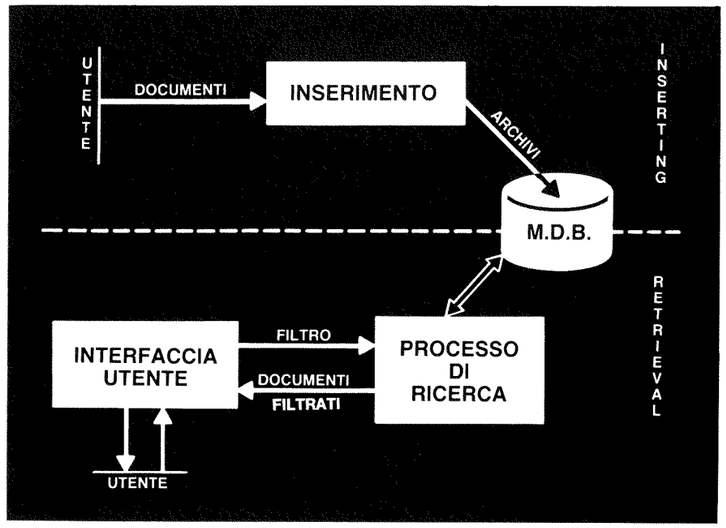


Fig. 1 - Architettura del sistema di archiviazione.

zione logica dei dati per potervi accedere, e deve saper descrivere esattamente cosa cerca, attraverso un opportuno linguaggio di interrogazione.

Portando un esempio tratto da un campo diverso, la gestione di una biblioteca, per entrare in possesso di un libro vi è normalmente una sola via: conoscere il nome dell'autore ed il titolo dell'opera. In questa organizzazione strutturata alcune richieste non possono essere soddisfatte, poichè di solito esistono schedari di ricerca per solo autore o per solo titolo. Se di un libro non si ricorda l'autore, ma si sa che ha la copertina rossa con un profilo bianco e che ha circa una sessantina di pagine, e quindi è sottile, queste informazioni non sono minimamente utili per la ricerca. Con una diversa organizzazione, invece, le stesse informazioni permetterebbero con ogni probabilità di recuperare immediatamente il libro. Anche in assenza di cataloghi costruiti su queste informazioni, se fosse consentito l'accesso ai locali dove sono conservati i libri, una paziente passeggiata tra gli scaffali, soffermando l'attenzione sui libri rossi e sottili, consentirebbe il buon esito della ricerca.

Chiaramente questo tipo di approccio è piuttosto inefficiente; se la biblioteca possiede molti libri, la ricerca potrebbe risultare assai lunga. Un miglioramento è ottenibile se l'organizzazione permette di utilizzare le informazioni possedute per ridurre lo spazio di ricerca. Se, come è ovvio, si ha un'idea dell'argomento del libro, per esempio si sa che tratta problemi di termodinamica, e se i libri sono raggruppati sugli scaffali per argomenti, si rivolgerà la ricerca agli scaffali dedicati alla fisica ed in particolare alla termodinamica.

L'esempio esposto si riferisce ad un sistema manuale, ma le stesse argomentazioni valgono anche per sistemi automatici: se l'accesso ad un archivio è permesso attraverso una chiave, solo usando la chiave si può ottenere l'informazione cercata.

In un multimedia data base le esigenze sono diverse. Esso va visto come una collezione poco strutturata di documenti multimodali in cui le modalità di accesso non sono prevedibili a priori: non è noto, cioè, come l'utente intenda estrarre le informazioni che desidera. Non si avrà quindi un metodo di accesso rigido ed univoco, ma un ambiente interattivo, che fornisce all'utente un insieme di nuove funzionalità che gli permettono di "navigare" nel data base fino al raggiungimento del proprio obiettivo. Questo ambiente interattivo deve inoltre possedere buone caratteristiche di facilità d'uso, in modo da essere accettato anche da utenti non specialisti, ai quali i multimedia data base hanno molto da offrire (si pensi al caso di una biblioteca elettronica).

2. Il metodo "filtering & browsing"

Il metodo *filtering & browsing* si propone di soddisfare le esigenze di accesso esposte nel paragrafo precedente, fornendo un metodo molto flessibile di accesso ad un multimedia data base. Il metodo è in linea di principio elementare e si compone di due fasi distinte: nella fase di *filtering*, viene effettuato un "filtraggio" del contenuto del data base per selezionare i documenti che soddisfano i requisiti richiesti dall'utente. Durante la fase di *browsing* (che letteralmente significa "sfogliare quà e là") si fornisce all'utente un opportuno ambiente interattivo che gli permette di spulciare tra i documenti filtrati finchè trova ciò che gli interessa oppure finchè si accorge di dover modificare i requisiti di filtering.

Vi è quindi una interazione tra le due fasi, poichè se il filtering seleziona i documenti da passare alla fase di browsing, è vero altresì che durante il browsing l'utente migliora la sua conoscenza dell'ambiente nel quale opera e può quindi essere in grado di definire meglio ciò che cerca restringendo o modificando i parametri di filtering.

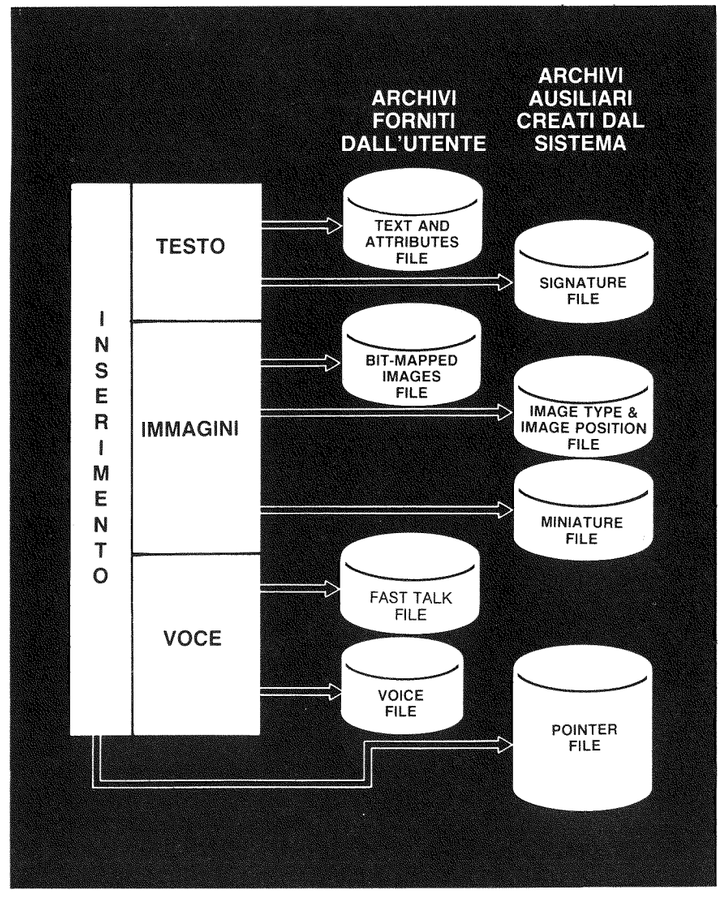
Questo approccio intende sfruttare al meglio le diverse capacità dell'uomo e della macchina. Da una parte infatti la macchina può essere molto efficace nel selezionare documenti sulla base dei valori di alcuni attributi (per esempio si richiedono documenti creati tra maggio e luglio del 1987 il cui mittente è Rossi); dall'altra l'uomo è più efficiente ed affidabile in una ricerca di tipo "visivo". Se l'utente ricorda che sul documento che sta cercando compare il logo di una certa azienda, piuttosto che la fotografia di una ragazza bionda, allo stato dell'arte è impossibile chiedere che la macchina sia in grado di reperirlo automaticamente: se anche si trattasse di qualcosa di più semplice, come la ricerca di un messaggio vocale che contenga una determinata parola, le attuali tecniche di *pattern recognition* renderebbero economicamente improponibile un tale tipo di ricerca su grandi data base.

Si può quindi affermare che con tale organizzazione la ricerca di un documento in un multimedia data base avviene in due fasi. Nel primo stadio lavora la macchina che, sulla base delle informazioni fornite dall'utente, filtra una parte di documenti. Nel secondo stadio lavora l'uomo che, adeguatamente supportato dalla macchina, passa in rassegna tutti o parte dei documenti proposti finchè trova ciò che va cercando oppure decide di ritornare alla fase di filtering.

L'idea in sè non è del tutto nuova: alcuni aspetti si possono ritrovare in altri metodi di accesso ad archivi. La tecnica delle chiavi confusionali [2], infatti, ha molti punti di contatto con questo metodo. In quel caso l'accesso avviene per chiave, però si ammette che l'utente la conosca solo con una certa approssimazione. Il sistema risponde alle interrogazioni selezionando un insieme di chiavi "vicine" (secondo una opportuna metrica) alla chiave richiesta; a questo punto l'utente può eseguire una ricerca manuale sulle chiavi selezionate oppure chiedere di cambiare i parametri di selezione. La tecnica delle chiavi confusionali non è collegata al metodo in esame, in quanto prevede solo accessi per chiave, però in essa è già presente l'idea che l'utente possa non conoscere con precisione ciò che sta cercando e quindi effettuare *da solo* una ricerca locale su un sottoinsieme di dati probabili.

Quanto finora esposto sul metodo filtering & browsing costituisce soltanto un insieme di principi generali a cui ispirarsi nel progettare sistemi di accesso a multimedia data base

Fig. 2 - Schema di inserimento di un documento multimodale.



basati su tale tecnica. I differenti ambienti applicativi impongono architetture diversificate, ed alla fantasia del progettista è lasciata la libertà di scegliere le strategie ed i meccanismi più opportuni con cui organizzare il sistema. Per fissare questi principi, nel prossimo paragrafo verrà presa in considerazione una possibile applicazione.

3. Una possibile applicazione

Si desidera ora analizzare una possibile architettura di accesso ad un multimedia data base con tecniche di filtering & browsing, con l'obiettivo di realizzare un archivio di documenti multimodali d'ufficio, costituiti cioè da:

- parte attributi (es: data di creazione, mittente)
- parte testo
- parte vocale (es: annotazioni registrate)
- parte immagini (es: disegni o foto)

Il problema consiste nel tratteggiare le caratteristiche di un sistema di archiviazione e di reperimento di tali documenti. Per fare ciò si possono seguire diverse vie.

Una prima tecnica è quella di etichettare ogni documento con un nome e memorizzare i documenti in una opportuna struttura dati che faciliti l'accesso per nome. In tale modo si avrebbe una buona efficienza ma, come già discusso in precedenza, comparirebbero problemi irrisolubili per la rigidità del metodo di accesso per sola chiave.

Una seconda possibilità consiste nell'archiviare sequenzialmente tutti i documenti, magari seguendo un determinato ordine (cronologico, alfabetico, ecc.), lasciando all'utente l'onere di scandire l'archivio per trovare ciò che cerca. In presenza di molti documenti questo approccio si rivela estremamente inefficiente, a meno che non ci sia un ordinamento privilegiato secondo cui l'utente indirizza sempre la sua ricerca. Nel caso in cui non si conosca l'attributo su cui è stato fatto l'ordinamento, si deve però ricorrere alla faticosa ricerca completa.

Un terzo approccio è quello di organizzare il sistema in modo che la ricerca possa essere effettuata sulla base del valore di alcuni attributi, attraverso un *query language* in cui si specificano le condizioni cui gli attributi del documento debbono sottostare. Per avvicinarsi alle esigenze dell'utente, il sistema dovrebbe permettere di specificare soltanto gli attributi richiesti.

Questo metodo va tanto meglio quanto più si riesce ad individuare un insieme di attributi che rappresentino adeguatamente il documento e che siano mnemonici, cioè cui l'utente generalmente ricorra per le sue ricerche. Rimane comunque il problema della rigidità evolutiva: può accadere che in futuro nuovi documenti richiedano per l'identificazione attributi non previsti in fase di progetto.

Una quarta possibilità è quella offerta dalla ricerca di un *pattern*. Questo metodo è oggi proponibile solo per la ricerca di porzioni di testo, in quanto, come già anticipato, per le immagini e la voce le tecniche di *pattern recognition* non sono ancora affidabili, nè tantomeno economiche. Per quanto riguarda i testi, invece, esistono algoritmi abbastanza efficienti che possono dare risultati validi.

I quattro approcci finora presentati sono per lo più orientati alla fase di filtering, in quanto descrivono un tipo di lavoro che può essere efficacemente svolto dalla macchina.

Un quinto approccio all'operazione di retrieval, che si appoggia invece all'uomo, è quello di cercare di incapsulare le caratteristiche del documento in termini di entità astratte che siano di facile interpretazione da parte dell'utente. Tali entità possono essere sia delle immagini che dei messaggi vocali che permettono di riconoscere rapidamente il documento rappresentato. Si tratta quindi di un tipico supporto alla fase di browsing, in cui vengono sfruttate le capacità di selezione umane.

Naturalmente la scelta del tipo di rappresentazione è soggettiva, però si possono adottare alcuni metodi abbastanza generali che garantiscono una buona funzionalità.

Uno di questi (che verrà adottato in seguito) è quello di costruire una rappresentazione visiva del documento che sarà detta *miniatura*.

La miniatura è un'immagine di dimensioni ridotte, così da dare solo un'impressione dell'aspetto del documento (o meglio della sua parte grafica e testuale), ma non ne permette la lettura. Una miniatura non è però semplicemente una riduzione in scala del documento originale, in quanto nel sistema non esiste una rappresentazione reale del documento, ma piuttosto un insieme di opportune strutture dati che ospitano le sue varie parti costituenti (voce, testo, immagini). Una miniatura è invece una costruzione ad hoc effettuata automaticamente al momento dell'inserimento del documento nel multimedia data base. Tale miniatura rimane associata al relativo documento per tutta la sua vita.

Un'altra soluzione che verrà adottata nel sistema in oggetto è quella dei *fast talk* (brevi messaggi vocali). Si tornerà comunque in seguito su questi punti.

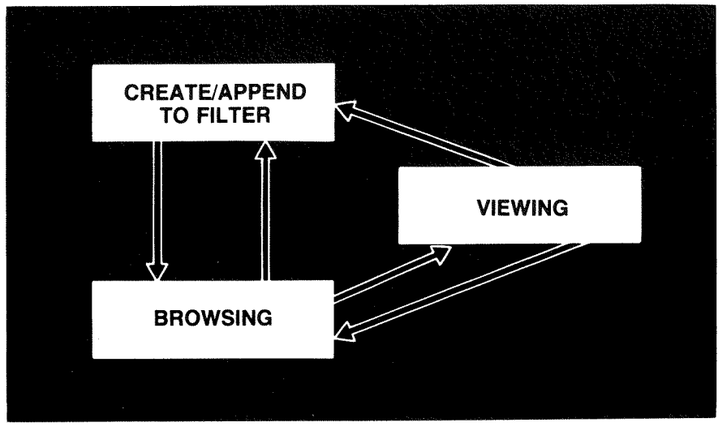
Ognuno dei metodi di retrieval finora esposti presenta alcuni vantaggi, e quindi per non rinunciare alla massima flessibilità che si vuole raggiungere, si adotterà una combinazione di tutte le tecniche al fine di sfruttare al meglio le informazioni che l'utente possiede per reperire il documento.

Dopo aver definito il campo di applicazione, nel prossimo paragrafo si passa a descrivere con maggior precisione l'organizzazione del sistema, così come proposta in [3], senza entrare però in dettagli implementativi.

4. Architettura del sistema

Un documento nel multimedia data base consiste di un unico identificatore e di un certo numero di attributi memorizzati in campi di lunghezza fissa che contengono data, mittente ed oggetto del documento. Vi sono inoltre dei segmenti di lunghezza variabile per il testo, le immagini e la voce. Le immagini pos-

Fig. 3 - Attività di retrieval svolte dall'utente.



sono consistere in grafici, tavole, intestazioni, istogrammi, diagrammi a torta, schemi, fotografie, etc e possono apparire in qualsiasi punto del documento. La parte vocale può essere costituita da commenti verbali relativi al documento, da annotazioni registrate per attirare l'attenzione del lettore, etc.

Nel sistema di archiviazione si possono individuare tre sottosistemi evidenziati in figura 1 da rettangoli:
1 - Inserimento di nuovi documenti
2 - Ricerca
3 - Colloquio con l'utente.

Il secondo ed il terzo sottosistema sono tra loro interagenti, in quanto attraverso l'interfaccia utente comunicano al processo di ricerca i parametri di filtering.

Tutti i documenti che soddisfano il filtro vengono quindi passati al processo che colloquia con l'utente perchè avvenga il browsing.

5. Inserimento

A differenza degli altri, il processo di inserimento è un processo autonomo, ma è molto importante in quanto è in questa fase che vengono creati i presupposti per il funzionamento di tutto il sistema di retrieval. Le informazioni create costituiscono rappresentazioni del contenuto di parte del documento e sono di quattro tipi:
- *Signature*
- *Miniature*
- *Image description*
- *Fast talk*

Il primo tipo viene creato per supportare l'accesso ai dati mediante *pattern* di testi. Viene creata un'apposita struttura dati, detta *signature file* (archivio di firme), che facilita l'operazione di *string-matching* in fase di filtering, utilizzando un opportuno algoritmo, che è di tipo approssimato, cioè risponde correttamente "quasi" sempre.

Gli errori sono comunque accettabili, perchè consistono in un maggior numero di risposte positive rispetto al reale.

Può quindi accadere che venga selezionato qualche documento di troppo (detto *false drop*), fatto, questo, non grave.

Il vantaggio di questo algoritmo consiste nella sua efficienza. Uno *string-matching* esatto sarebbe molto pesante dal punto di vista computazionale oppure dal punto di vista dello spazio occupato. I *signature file* sono invece di dimensioni modeste ed in fase di filtering la ricerca del *pattern* di testo avviene accedendo solo a questi archivi, e non al documento che potrebbe essere molto voluminoso.

Inoltre la ricerca è assai veloce perchè la maggior parte del lavoro viene fatta per la costruzione delle *signature,* quindi in fase di inserimento e non di retrieval.

Il secondo tipo di informazione che viene creata è detta *miniatura* ed è già stato introdotto in precedenza. Per costruire le miniature delle righe di testo vengono associate linee di spessore variabile (per rappresentare le lettere ascendenti e discendenti), mentre la sagoma del testo che si ottiene viene composta con la riduzione delle immagini bit-mapped. Si ottiene così una figura che ricorda da vicino il *layout* del documento originale e può essere utile in fase di browsing se l'utente ha qualche idea sulla struttura del documento.

Un'altra informazione che viene rilevata in fase di inserimento è la descrizione delle immagini che appaiono sul documento. Questa descrizione serve solo per individuare il tipo di immagine (grafico, diagramma a torta, istogramma, foto, ecc.) e la sua posizione all'interno del documento. Ciò potrà essere utile durante il retrieval in quanto l'utente potrà, per esempio, chiedere al processo di filtering di selezionare i documenti che presentano un istogramma nella seconda pagina in basso a destra.

La quarta ed ultima informazione è rappresentata dai *fast talk*. Questi sono brevi messaggi vocali di descrizione dei documenti, creati al momento dell'inserzione nel data base, oppure sono stralci del contenuto della parte vocale del documento. Questo strumento permette di allargare ulteriormente la banda di comunicazione fra uomo e macchina.

I *fast talk* verranno utilizzati in fase di browsing insieme alle miniature per facilitare l'identificazione del documento.

Riepilogando, durante la fase di inserimento dei documenti nel multimedia data base l'utente deve forni-

re quattro archivi che contengono rispettivamente:

- la parte attributi ed il testo
- le immagini
- la voce
- il *fast talk* (che non fa parte del documento ma verrà comunque memorizzato insieme ad esso).

Sulla base di questi quattro archivi, il sistema ne crea automaticamente altri tre di informazioni ausiliarie per l'accesso, memorizzando dunque il documento in un totale di sette archivi diversi, ma correlati, che possono essere visti come *segmenti* del multimedia data base. Vi è infine un ulteriore *pointer file* che contiene le informazioni per ricostruire il documento nella sua interezza. Lo schema degli archivi in fase di inserimento è rappresentato in figura 2.

6. Interfaccia utente e processo di ricerca

L'interfaccia utente è la componente del sistema che si occupa del colloquio con l'utente. Si tratta di un ambiente interattivo, in cui l'utente può svolgere tre tipi di attività cui corrispondono altrettanti modi operativi. Le attività utente, insieme alle transazioni ammesse, sono rappresentate in figura 3.

Passando all'analisi delle tre attività, si rileva quanto segue.

1) Creazione o modifica del "filtro"

Si definisce *filtro* quell'insieme di specifiche a cui il sistema fa riferimento per selezionare i documenti da proporre in output (vedi figura 1). Nel creare il filtro l'utente dispone di molta libertà. Non gli viene richiesta nessuna informazione particolare, ma gli si permette di specificare soltanto ciò che sa. Si può quindi richiedere che la selezione venga fatta in base a:

- restrizioni su particolari attributi del documento (es: data creazione precedente all'agosto 1987);
- ricerca di un certo *pattern* di testo (es: documento il cui testo contenga la stringa "bulloni cromati".
  È possibile specificare condizioni più complesse, come congiunzioni o disgiunzioni e negazioni)
- si può specificare se il documento possiede o meno una parte vocale
- si può specificare se ci sono o meno immagini, ed in caso affermativo si può precisarne il numero, la posizione nel documento ed il tipo (istogramma, schema, foto, ecc.).

Essendo ammessa qualsiasi combinazione di queste richieste, si ottiene uno strumento estremamente flessibile. Quando l'utente pensa di aver completato la descrizione del filtro, può passare alla fase di browsing. Il processo di ricerca si avvia in modo trasparente, e comincia a selezionare i documenti che soddisfano i requisiti spedendone le relative astrazioni (*miniature* e *fast talk*) all'interfaccia utente, attraverso la quale verrà effettuato il browsing.

2) Browsing

L'utente può scandire le miniature ed ascoltare i *fast talk* dei documenti selezionati al fine di individuare quello cercato. Le miniature scorrono sullo schermo e l'utente può fissare velocità e verso di scorrimento (avanti o indietro). I *fast talk* vengono fatti ascoltare (opzionalmente) in corrispondenza alla relativa miniatura.

Se durante questa fase l'utente decide di cambiare le specifiche di filtering può farlo liberamente, e passare quindi nuovamente all'attività di creazione/modifica del filtro. Se invece individua un documento che gli interessa può chiedere di passare all'attività di *visualizzazione* o *viewing*.

3) Visualizzazione

L'utente può chiedere che il documento selezionato durante il browsing venga riprodotto su video in forma conforme all'originale (le miniature sono soltanto sagome intuitive), e che la parte vocale venga eseguita. Se il documento che appare è proprio quello cercato, l'utente può ora consultarlo, altrimenti può decidere di continuare il browsing oppure di modificare il filtro.

7. Conclusione

In questa rassegna delle caratteristiche architetturali di un sistema di ricerca in data base multimodali basato sul metodo di filtering & browsing non si sono affrontate le problematiche implementative, per cui le prestazioni ottenibili da un tale sistema restano una questione da approfondire. Le caratteristiche salienti dal punto di vista dell'interazione tra uomo e macchina sono però molto promettenti e giustificano gli sforzi che molti ricercatori stanno dedicando all'argomento.

Bibliografia

[1] L. FELICIAN, T. TIZIANEL: *"Dai dati ai documenti multimedia"*, Quaderni di informatica, 31, 1987.

[2] R. BORGINI, L. FELICIAN: *"Data Processing as a marketing support: the experience of GENERALI"*, Proc. 10. Semaine Internationale de l'Informatique dans l'assurance, CAPA, Paris 1986.

[3] D. TSICHRITZIS et al.: *A Multimedia Office Filing System"*, Proc. VLDB 9th Conference, Firenze 1983.

[4] AMIHAI MOTRO: *"Browsing in a Loosely Structured Database"*, Proc. ACM SIGMOD, Boston 1984.

[5] S. CHRISTODOULAKIS et al.: *"The Multimedia Object Presentation of MINOS: A Simmetric approach"*, Proc. ACM SIGMOD, Washington 1986.

[6] J. CONKLIN, *Hypertext: An introduction and survey*, IEEE Computer, Sept. 1987.

Tecniche di stampa per l'informatica individuale

GIAN CARLO GATTI

Consulente sistemi di stampa

1. Introduzione

In questi ultimi anni, in seguito alla grande diffusione dei Personal Computer (PC) usati in attività di elaborazione dati e di editoria da tavolo (Desk Top Publishing), le stampanti si sono molto evolute presentando caratteristiche e prestazioni assai interessanti e di valore che le differenziano sensibilmente dalle stampanti costruite all'inizio degli anni '80.

Questo articolo presenta una analisi sintetica delle moderne tecnologie di stampa con specifico riferimento all'utilizzazione nelle stampanti collegate ai PC. L'analisi riguarda i principi tecnologici delle varie soluzioni e le loro caratteristiche funzionali e applicative, incluso un confronto dei costi operativi.

2. Tipologia delle applicazioni

Le varie tecniche di stampa possono essere associate a differenti aree applicative. Per una classificazione delle varie soluzioni tecnologiche si ricorre si solito ad uno schema a tre livelli: famiglia, categoria, classe.

La famiglia distingue la tecnica di stampa, che può essere ad impatto (Impact Printer) e non ad impatto (Non Impact Printer); la categoria indica il metodo di stampa, che è o seriale (un carattere per volta) o parallelo (una linea di caratteri per volta); la classe individua il tipo di carattere da stampare, che può essere ad immagine intera (Solid Font) o per punti (Dots Matrix).

Le velocità di scrittura sono espresse in CPS (Caratteri Per Secondo) per le seriali ed in LPM (Linee di caratteri Per Minuto) per le parallele. Per le stampanti a pagina la velocità è in PPM (Pagine Per Minuto) ed è indipendente dal numero o linee di caratteri per pagina.

Con queste precisazioni si può meglio interpretare la tabella di fig. 1 che elenca le tecnologie di stampa sviluppate in questi ultimi due decenni, con le relative caratteristiche.

Nell'ultimo decennio, in particolare, l'evoluzione delle tecnologie di stampa è stata rapida e tumultuosa. Infatti sono state sviluppate molte tecniche, alcune sono progredite migliorando le loro prestazioni, altre hanno avuto vita breve o sono state relegate in piccole aree applicative (nicchie), dove l'alto costo di stampa, visto il particolare e quantitativamente piccolo lavoro da svolgere, non è ritenuto un "killer" della tecnologia, ossia un fattore che la mette fuori causa.

Grazie agli sviluppi dell'informatica, l'arte della stampa è entrata negli uffici e da poco tempo anche nelle abitazioni private. La rapida espansione dei Personal Computer ha spronato i costruttori di stampanti a studiare e a sviluppare tecnologie più sofisticate atte a soddisfare le richieste del mercato.

Le stampanti collegate ai PC sono una valida dimostrazione di quanto oggi queste macchine possono fare. Esse stampano testi con vari stili (Font) e caratteri con una eccellente qualità di stampa, di livello simile a quello tipografico. Inoltre, grazie alla loro flessibilità, possono stampare grafici assai complessi e riprodurre disegni e fotografie da inserire nei testi.

Come esempio sono qui citate brevemente alcune aree applicative oggi in rapida espansione:

a) Studi professionali, piccole aziende:
   - contabilità generale;
   - contabilità magazzino;
   - IVA, paghe, stipendi;
   - calcoli, statistiche.

b) Ufficio piccolo/medio:
   - testi con alta qualità di stampa;
   - grafici non complessi.

c) Desk Top Publishing (editoria da tavolo):
   - testi con stili del carattere diversi;
   - fotografie incorporate nel testo;
   - disegni e grafici complessi;
   - prodotti tipografici finiti, per esempio cataloghi.

Tramite la tastiera del PC si utilizzano tutte le funzioni del sistema: linee orizzontali, verticali, angoli, cerchi, ingrandimenti o riduzioni di disegni, grafici, caratteri diversi, ecc. per costruire sul video, ed eventualmente correggere, il documento che può essere, per esempio, parte in testo, in grafica e con una fotografia inserita. La foto viene "letta" da un lettore d'immagini (scanner) collegato al PC.

TECNICHE DI STAMPA	FAMIGLIA		CATEGORIA		CLASSE		VELOCITÀ (max)			AMBIENTE APPLICATIVO
	IP	NIP	SER.	PAR.	SOLID FONT	DOT MATRIX	CPS	LPM	PPM	
TAMBURO	X			X	X			1.500		CENTRO EDP
NASTRO/CATENA	X			X	X			2.000		CENTRO EDP
PETTINE	X			X		X		1.100		CENTRO EDP
CESTELLO	X		X		X		30			UFFICIO
CILINDRO	X		X		X		15			UFFICIO
SFERA	X		X		X		20			UFFICIO
MARGHERITA	X		X		X		60			UFFICIO
AGHI	X		X			X	500			UFFICIO/CASA
TERMICA DIRETTA		X	X			X	15			UFFICIO/CASA
TERMICA CON TRASFERIM.INCHIOSTRO		X	X			X	180			UFFICIO/CASA
GETTO INCHIOSTRO		X	X			X	300			UFFICIO
ELETTROSTATICA		X		X		X		1.500		CENTRO EDP
SCARICA ELETTRICA		X		X		X		1.000		UFFICIO
ELETTROFOTOGRAFICA		X	X			X			150	UFFICIO/EDIT. DA TAVOLO
MAGNETOGRAFICA		X		X		X			70	CENTRO EDP
IONOGRAFICA		X		X		X			60	UFFICIO

Fig. 1 - Quadro complessivo delle tecniche di stampa.

È così possibile produrre in casa, con una veste tipografica di qualità, testi, libretti, coupon, cataloghi, ecc in tempo reale, ovvero non risulta più indispensabile rivolgersi alle tipografie, con vantaggi economici, di di riservatezza e di tempo. In Europa, Italia compresa, stanno nascendo dei centri di stampa rapida (Quick Printing) per dare appunto questo servizio alle piccole aziende che non hanno la convenienza economica di acquistare un sistema di Desk Top Publishing, per la limitata quantità di stampa di cui hanno bisogno.

La composizione dell'immagine avviene per punti secondo una determinata matrice o reticolo (numero di punti per inch^2). Tutta l'immagine, ed i vari programmi d'impaginazione, normalmente sono registrate su un disco magnetico inserito nel PC.

Terminata la composizione dell'immagine, si ordina alla stampante di scrivere tutto il contenuto del video (che è registrato nella memoria del PC) su un foglio di carta.

La tecnologia ad aghi è largamente utilizzata per l'applicazione "a" e "b" mentre per la "c", molto più complessa, la tecnologia elettrofotografica con scrittura laser è la sola impiegata.

Nei prossimi capitoli saranno descritte le più importanti tecnologie di stampa ad impatto e non.

3. Stampa ad impatto

Questo tipo di stampa è meccanica. Gli elementi di scrittura, come aghi o caratteri coniati su vari supporti, sono portati, tramite mezzi meccanici, a battere contro la carta con interposizione di un nastro inchiostrato.

Questa tecnica, che ha nella macchina per scrivere un classico esempio, è l'unica a generare un originale e diverse sue copie contemporaneamente. Gli elementi di scrittura provocano una pressione contro il nastro inchiostrato ed i fogli di carta carbone, sufficiente a staccare da essi particelle di inchiostro, le quali si depositano sulla carta formando il carattere.

La scrittura ad impatto può essere seriale o parallela.

Il carattere può essere stampato in modo pieno (Solid Font) cioè come il tipografico o per punti (Dots Matrix). Con la tecnica a punti il carattere è costruito secondo una determinata matrice.

Vi sono matrici più o meno dense di punti per soddisfare esigenze di scrittura diverse.

La stampa ad impatto, come si intuisce, produce rumore che però con adeguati accorgimenti si può ridurre ad un livello accettabile.

Fig. 2 - Tecnica di stampa a sfera.

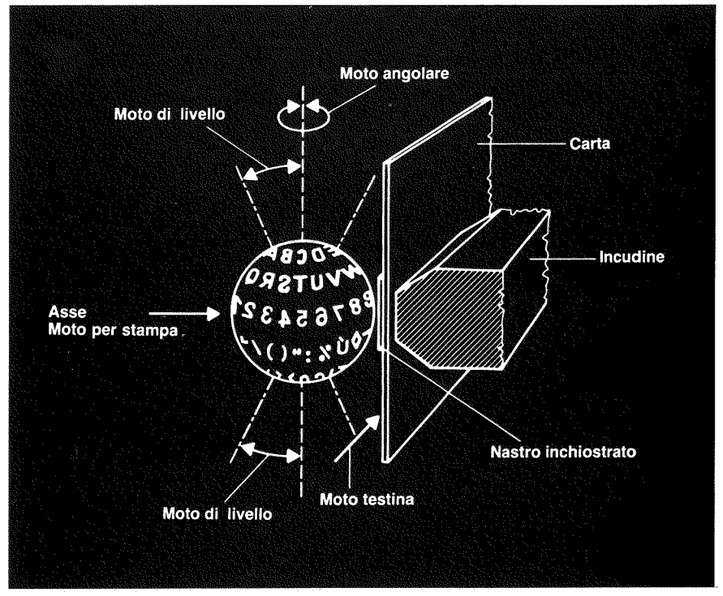
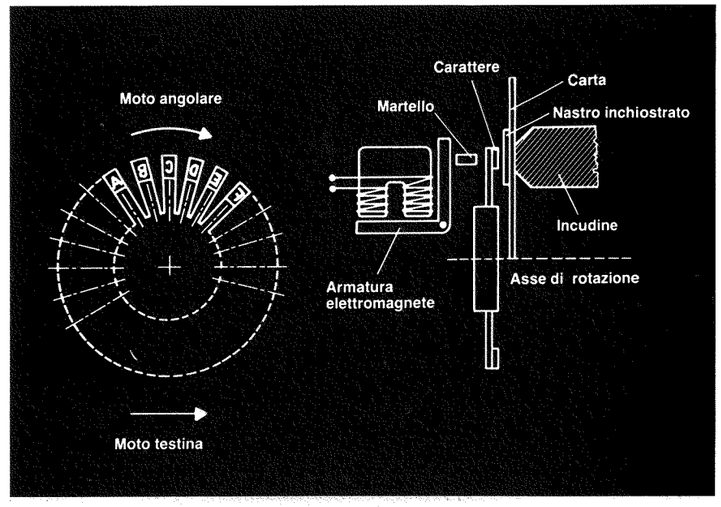


Fig. 3 - Tecnica di stampa a margherita.



Negli ultimi dieci anni si sono diffuse, maggiormente, tre tecnologie ad impatto: sfera, magnetica e aghi.

Queste tre tecnologie saranno oggetto nei prossimi paragrafi, mentre delle altre tecnologie seriali (cestello e cilindro) non verrà fatto cenno poichè trattasi di tecniche in disuso. Anche le tecniche di stampa ad impatto parallelo (tipo: nastro, catena, tamburo e pettine) non saranno prese in esame, poichè queste stampanti non sono usate in connessione con Personal Computer, ma in configurazioni di elaboratori più potenti.

3.1 Sfera

Con questa tecnica il carattere è del tipo pieno e tutto il set (totale numero dei caratteri disponibili) è contenuto sulla superficie di una piccola sfera in acciaio o plastica. In fig. 2 è illustrato uno schema di stampa a sfera.

I caratteri sono coniati nello stile voluto e posizionati sull'intersecazione dei "meridiani" e "paralleli". Il carattere da stampare è selezionato con opportuni mezzi elettromeccanici che agiscono sul moto angolare (meridiani) e sul moto di livello (paralleli). Terminata la selezione, il carattere da stampare si troverà sull'"equatore" (asse moto stampa). Altri mezzi porteranno la sfera a battere contro il nastro inchiostrato e la carta.

La scrittura è seriale e la velocità massima è di 20 CPS. La testa di scrittura è montata su un carrellino mobile e traslata lungo la linea di stampa. Si possono ottenere scritture con stili di caratteri diversi: si sostituisce, con semplice operazione manuale, la sfera. Buona la qualità di stampa.

3.2 Margherita

Anche con questa tecnica il carattere è pieno ed è coniato su supporti in acciaio od in plastica. Esso è montato su delle lamine elastiche (petali) ruotanti su un asse. In Fig. 3 è mostrato un esempio di stampa a margherita.

Questa ruota portacaratteri ed il suo equipaggiamento (motore elettrico, elettromagnete con martello, sincronismo di sparo martello, nastro inchiostrato) sono montati su un carrello in movimento lungo la linea di stampa. La ruota margherita viene messa in rotazione da un motore elettrico ed il suo moto angolare è controllato da un dispositivo di sincronismo che segnala il raggiungimento del carattere selezionato.

In coincidenza con la posizione di stampa lungo la linea (spostamento del carrello pure esso controllato) viene energizzato, tramite l'elettromagnete, il martello il quale spinge il petalo contro il nastro inchiostrato formando così sulla carta il carattere.

La velocità di stampa massima è di 60 CPS. La ruota margherita si sostituisce manualmente con altre aventi caratteri e stili diversi. Con questa tecnica si ottiene una ottima qualità di stampa.

3.3 Aghi

Il carattere, a differenza delle due tecniche precedenti, è formato sulla carta da una serie più o meno grande di piccoli punti, tutti contenuti in un'area detta matrice.

L'elemento di stampa ha la forma di un ago, per esempio di 0,35 mm di diametro, ed è spinto ad alta velocità da mezzi opportuni contro il nastro inchiostrato e la carta.

Questa tecnica ad aghi permette di avere organi meccanici molto piccoli e quindi masse ridotte. Le velocità ad inerzie del cinematismo di stampa risultano così agevolmente controllate. La costruzione del carattere fatta per punti, uguali fra loro, permette, a differenza della stampa del carattere pieno, un migliore dosaggio dell'energia conferita poichè essa è uguale per ogni punto. Nel carattere pieno si hanno invece squilibri di energia conferita all'elemento di stampa (sfera o margherita) poichè vi sono delle sensibili differenze di superfici tra caratteri: per esempio tra una "B" e un punto che è il carattere con la superficie più piccola.

Pertanto, con la tecnica ad aghi, è necessaria una minore energia a tutto vantaggio del rumore acustico prodotto dall'impatto dell'elemento di scrittura.

In fig. 4 è raffigurato lo schema classico di stampa ad aghi. Lo schema di principio di scrittura, per esempio a 9 aghi, è illustrato con un solo elettromagnete di comando ed un ago. Gli altri 8 uguali che completano la testina sono appena tracciati.

Ciascun elettromagnete, tramite la propria armatura o paletta, ricevuto un apposito comando sincrono con la posizione della testina di scrittura, energizza il proprio ago il quale, scorrendo in apposite guide, si comporta come un proiettile. Quando la punta dell'ago urta contro il rullo o l'incudine comprime il nastro e la carta, generando un punto su di essa.

Anche in questa tecnica, essendo seriale, la testina di scrittura è montata su un carrellino che trasla lungo la linea di stampa. A differenza delle tecniche a sfera e margherita, la stampa viene eseguita con carrello in movimento.

3.3.1 Costruzione del carattere a punti

Nella matrice a punti si indica, generalmente con il primo numero, la densità dei punti stampabili in orizzontale (H) per inch (25,4 mm) e con il secondo la densità verticale (V) che spesso coincide con il numero di aghi od elementi di scrittura. Vi sono matrici più o meno dense di punti (DPI = Dots Per Inch) con cui è possibile ottenere testi con caratteri diversi senza la necessità di sostituire gli elementi di stampa. È da sottolineare che questa tecnica permette la costruzione di diagrammi e figure che praticamente non è possibile realizzare con la scrittura a carattere pieno.

Le matrici più comuni sono di 63 o 81 punti: 7/9 righe orizzontali (H) e 9 colonne verticali (V). Il set grafico si realizza all'interno della matrice o reticolo scegliendo opportunamente una serie di punti per colonne e per righe contenuti nella memoria del generatore di caratteri. Il metodo di costruzione del carattere a punti è illustrato in fig. 5.

Lo spostamento lungo l'asse di scrittura del carrellino porta testina ad aghi è controllato da uno o più sensori che "leggono" delle tacche o fori posti sul disco di sincronismo in modo da segnalare ad ogni istante la posizione degli aghi lungo la linea di stampa.

Allorchè la fila degli aghi è in coincidenza con la prima delle 7 colonne della matrice del carattere da stampare, il generatore di caratteri invia un segnale elettrico agli elettromagneti per energizzare gli aghi 4, 5, 6, 7 e successivamente l'ago 3 sulla colonna 2, gli aghi 2, 5 sulla colonna 3, l'ago 1 sulla colonna 4 , gli aghi 2, 5 sulla colonna 5, l'ago 3 sulla colonna 6 ed infine gli aghi 4, 5, 6, 7 sulla colonna 7.

Il questa maniera il carattere A è stampato con sette fasi di scrittura. Gli aghi 8, 9 sono utilizzati per i tratti chiamati "discendenti" (per esempio carattere p minuscolo o sottolineature).

3.3.2 Risoluzione del carattere a punti

Per ottenere una maggiore risoluzione del carattere ovvero migliorare la qualità di stampa, si aumenta la densità della matrice in modo da avvicinare o sovrapporre parzialmente i punti fra loro. Infatti con matrici molto dense l'immagine risultante del carattere appare fatta come un tratto continuo e se i punti sono molto piccoli in diametro il bordo dell'immagine appare netto. In altre parole con matrici adatte, punti piccoli e particolari metodi di stampa si possono ottenere caratteri molto simili a quelli tipografici. Come appare ovvio, per avere una buona qualità di stampa occorre aumentare il numero degli aghi o punti in verticale ed i punti in orizzontale.

In fig. 6 sono illustrate due geometrie di testine ed aghi e due tipi di densità orizzontale.

Con la geometria A la densità verticale è alquanto bassa. La si può aumentare in due passate arrivando così a 18 punti verticali: si stampa metà dei punti in una prima passata poi si attiva il moto della carta per mezzo passo e si effettua una seconda passata di stampa.

La soluzione B (2 file di aghi) è la migliore: non richiede doppia passata e ha una maggiore velocità e densità rispetto ad A.

In C sono mostrati i punti orizzontali: la testina ad aghi si posiziona 7 o 9 volte nella larghezza della matrice per stampare, secondo la geometria o tipo della stessa, più o meno punti. I circoli indicano tutte le posizioni del punto mentre i trattini verticali mostrano i punti usati dagli aghi verticali adiacenti (questo è molto importante per le "diagonali"; vedi per esempio il carattere A) e possono essere scritti finchè lo stesso ago non è attivato due volte di seguito.

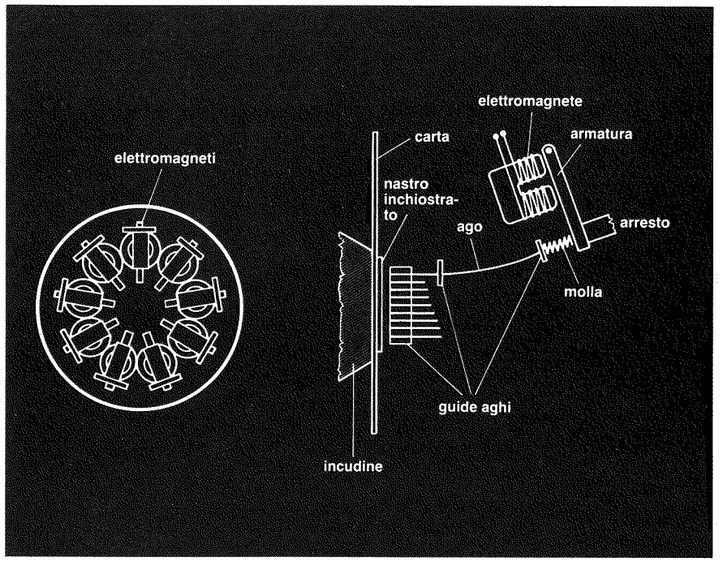
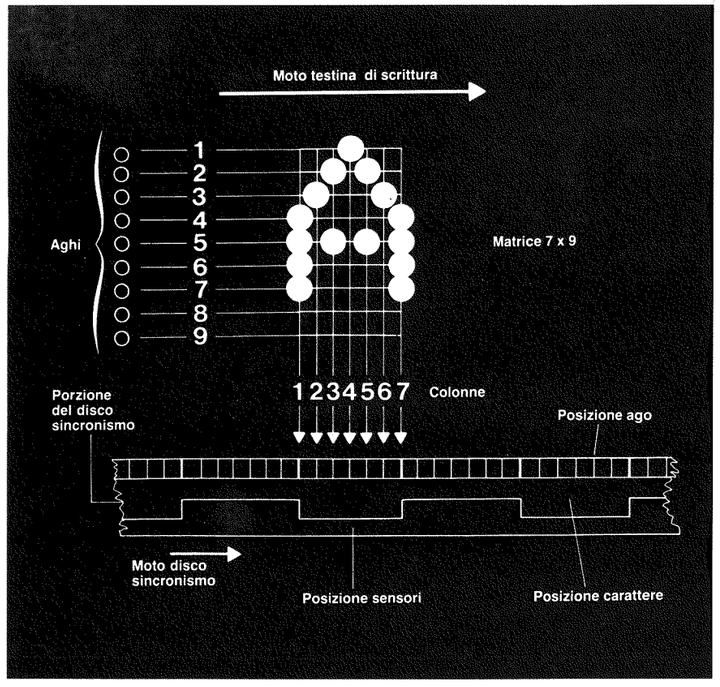


Fig. 4 - Tecnica di stampa ad aghi.

Fig. 5 - Costruzione del carattere a punti.



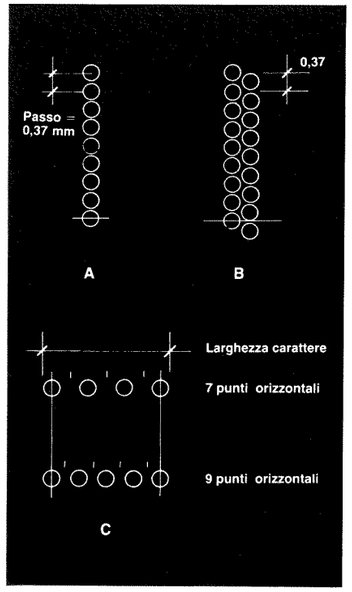


Fig. 6 - Geometrie e matrici per testine ad aghi.

In fig. 7 sono mostrati vari esempi di scrittura con densità H e V crescenti. Aumentando la densità dei punti ed ottimizzandone la posizione si ottiene un disegno del carattere simile al pieno raffigurato in figura.

La velocita di stampa è strettamente legata alla densità della matrice. A frequenza di scrittura (singoli punti) uguale, risulta più veloce la stampa di un carattere avente la risoluzione meno alta. In altri termini quanto è maggiore la densità della matrice, sempre a pari velocità dei mezzi di scrittura, tanto proporzionalmente minore è la velocità di stampa. La velocità di scrittura massima raggiungibile con la tecnologia ad aghi è di circa 250 CPS con matrice 7x9 e di circa 500 CPS con la nuova tecnica a magnete permanente (vedi par. 6.1).

In fig. 8 è mostrata una fotografia della testina di scrittura Honeywell Bull Italia a 9 aghi a magnete permanente.

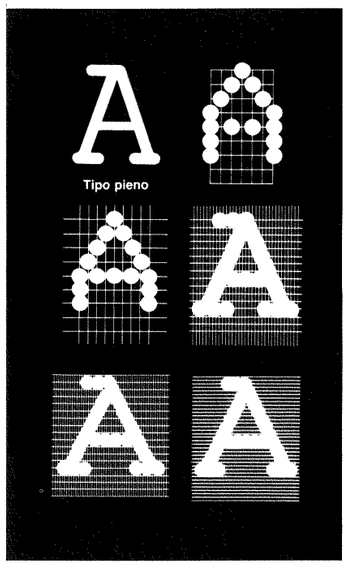


Fig. 7 - Esempi di scrittura a punti con densità crescenti.

4. Stampa a non impatto

Le tecnologie di stampa a non impatto (NIP) utilizzano mezzi di scrittura che sfruttano principi ottici, magnetici, termici ed elettrici. In questo tipo di stampa i mezzi di scrittura hanno poche parti in movimento e la "battuta" è eliminata riducendo così sensibilmente il rumore acustico.

La velocità di stampa è assai più alta rispetto alla tecnica ad impatto ma non si possono ottenere copie dell'originale contemporaneamente.

La scrittura NIP è sempre fatta per punti utilizzando matrici che in alcuni casi raggiungono densità molto elevate, ottenendo così un'alta qualità di stampa. La scrittura è seriale e le velocità sono espresse in CPS o PPM.

All'inizio degli anni '80 la stampa a non impatto si è sviluppata rapidamente soprattutto per la sua silenziosità di stampa e per la migliore qualità ottenuta. Varie tecnologie sono state generate, le più significative sono:

- getto d'inchiostro a domanda (Ink Jet Drop On Demand);
- termica con trasferimento d'inchiostro (Thermal Ink Transfer);
- elettrofotografica con scrittura seriale a laser.

Anche per queste tre tecnologie saranno illustrati nei prossimi paragrafi i principi di stampa e le prestazioni.

Per le altre tecnologie NIP (termica diretta, elettrostatica, scarica elettrica, magnetografica e ionografica) verrà omessa la descrizione poichè esse non sono molto diffuse ovvero sono usate in nicchie particolari di applicazioni.

4.1 Getto d'inchiostro

La scrittura con la tecnica ink jet viene eseguita inviando in modo selettivo delle gocce d'inchiostro contro la carta. La costruzione del carattere è fatta per punti tutti contenuti dentro una determinata matrice.

Due tecniche sono state sviluppate in questi due ultimi decenni: getto di gocce d'inchiostro continuo e getto di una goccia su domanda (DOD). L'inchiostro impiegato è liquido ed essica entro breve tempo sulla carta.

La descrizione del funzionamento della testina ink jet sarà fatta solo per la tecnica DOD seriale perchè è la più diffusa.

La testina di scrittura, posta su un carrellino mobile che si sposta lungo la linea di stampa, ha un adeguato numero di ugelli, per esempio 9 sul piano verticale, dai quali a comando esce in velocità una goccia d'inchiostro verso la carta che dista circa 1 mm. Si può pensare che le gocce d'inchiostro inviate sulla carta dagli ugelli siano posizionate come le particelle d'inchiostro trasferite dagli aghi di una testina seriale ad impatto.

La goccia è spinta ad uscire dall'ugello da mezzi come trasduttori piezoelettrici.

Il trasduttore piezoelettrico è posto nella camera di eiezione, sempre piena d'inchiostro, in modo tale che quando viene eccitato da un segnale

elettrico, proveniente dal generatore di caratteri (vedi par. 3.3.1), provoca, per azione meccanica, un piccolo restringimento della sezione della camera. Questa variazione genera un aumento di pressione con la formazione di un'"onda" che spinge fuori dall'ugello una goccia d'inchiostro. In fig. 9 è illustrato uno schema di principio di scrittura ink jet DOD seriale.

La costruzione del carattere a punti è uguale al metodo descritto per la testina ad aghi, tenendo conto che un ugello equivale ad un ago. La velocità di scrittura può arrivare sino a 300 CPS. Ottima la qualità di stampa se viene usato un foglio di carta pregiata: patinata e poco assorbente.

4.2 Termica con trasferimento d'inchiostro

Anche con questa tecnologia (Thermal Ink Transfer) la scrittura è seriale e per punti. Questa tecnica sfrutta l'energia termica per fondere un piccolo strato d'inchiostro posto su un sottile nastro termico che è a contatto con la carta.

Il nastro termico è costruito con un minimo di 2 strati (1 di base e l'altro con inchiostro) ed un massimo di 4 (1 di base, 2 conduttori di elettricità ed 1 con inchiostro) che rilasciano, tramite calore, generato selettivamente da una testina termica, punti d'inchiostro sulla carta.

La testina termica, posta su un carrello mobile che si muove lungo la linea di stampa, è composta da una serie di microresistori allineati verticalmente (geometrie, per esempio, uguali alle testine ad aghi), che sono attivati dai segnali elettrici provenienti dal generatore di caratteri (vedi fig. 10).

Il calore generato dai microresistori attraversa il sottile strato di base, a contatto con la testina, fondendo l'inchiostro. La costruzione del carattere ricalca il metodo usato per la tecnologia ad aghi ed ink jet DOD.

La massima velocità di scrittura è nell'ordine dei 180 CPS con un'ottima qualità di stampa.

4.3 Elettrofotografica con scrittura laser.

La scrittura elettrofotografica seriale a laser è quella, tra le stampanti NIP, che da le migliori prestazioni: velocità, eccellente qualità di stampa e silenziosità.

Il laser a diodo è un dispositivo semiconduttore che emette un raggio di luce monocromatico e coerente. Il raggio emesso deve essere modulato per formare un treno di piccoli punti luminosi in successione, secondo la matrice del carattere da stampare. Il generatore di caratteri,

Fig. 8 - Testina di scrittura a 9 aghi, con magnete permanente (progetto Honeywell Bull Italia)



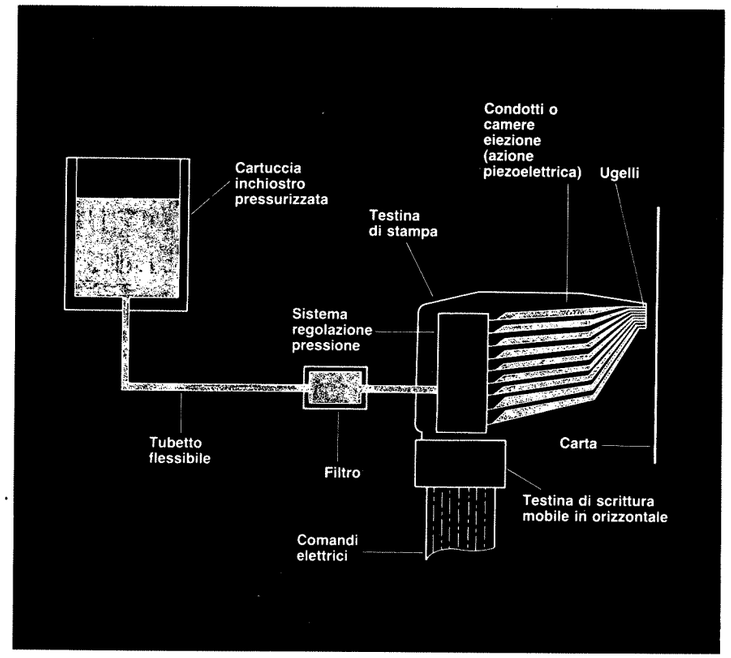
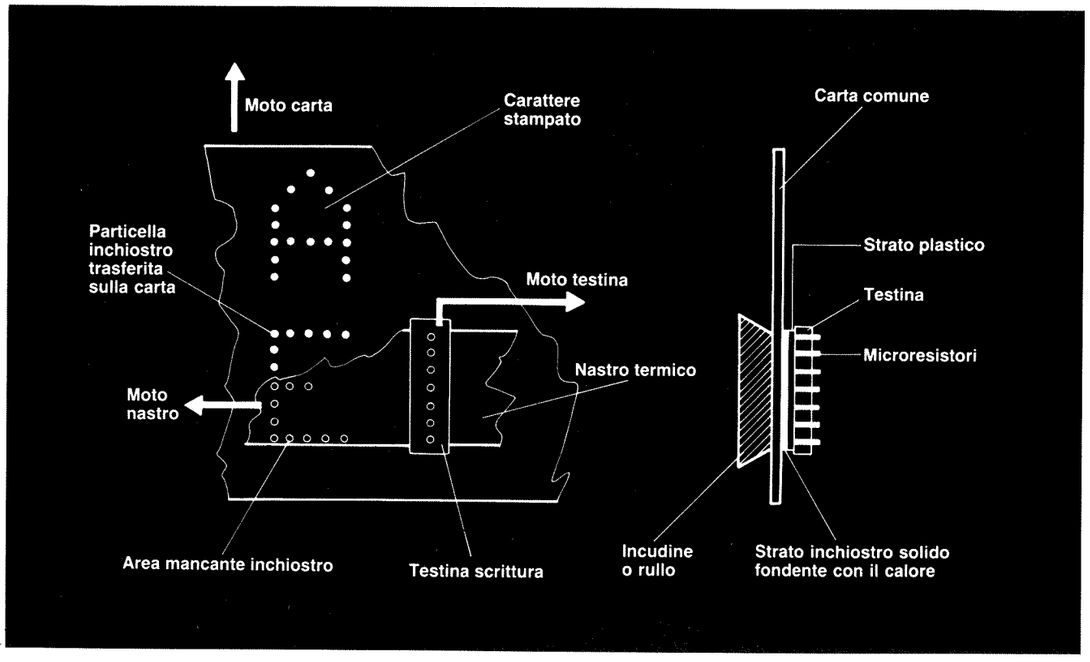


Fig. 9 - Tecnica di scrittura a getto di inchiostro seriale, tipo DOD.

Fig. 10 - Tecnica di scrittura termica seriale, con trasferimento di inchiostro.



che ricordiamo è un dispositivo elettronico dove sono mappati tutti i punti del carattere, controlla il diodo laser per interrompere il raggio laddove esso non deve scrivere.
Il treno di puntini luminosi viene riflesso da uno specchio rotante (scansione orizzontale) in modo da ottenere una riga di puntini lungo la linea di stampa. Con opportune lenti ottiche si possono avere punti con diametri da 0,05 a 0,12 mm che sono

valori da quattro a due volte più piccoli dei punti ottenibili con i mezzi di scrittura con tecnologie ad aghi, ugelli, resistori. Pertanto, con adeguate frequenze di emissione dei puntini luminosi da parte del diodo laser, legate alla velocità di rotazione dello specchio e del tamburo (sul quale si forma l'immagine) e con un diametro dei punti sufficientemente piccolo, si possono realizzare matrici molto dense atte a creare immagini, testi e grafici di alta qualità.

Il processo di stampa elettrofotografico si basa sulla formazione di una immagine latente (invisibile) composta da aree caricate elettrostaticamente (bianco) ed aree senza carica (nero), giacenti sulla superficie di una materiale sensibile alla luce (fotoconduttore). L'immagine latente così composta sarà successivamente resa visibile facendo aderire, nelle aree scariche, delle particelle d'inchiostro (toner) che saranno poi trasferite su un foglio di carta e fissate ad esso.

Questo processo elettrofotografico utilizza le proprietà del materiale fotoconduttore (per esempio Selenio). Esso infatti è fortemente isolante al buio e diventa un discreto conduttore quando è esposto alla luce. Detto materiale, prima della scrittura col fascio laser, viene, con mezzi opportuni, caricato elettrostaticamente in modo uniforme. Allorchè si illumina con il raggio laser un punto del fotoconduttore, la scarica elettrostatica locale si disperde, mentre nell'area non illuminata essa rimane.

Il processo di stampa elettrofotografico può essere suddiviso in 6 stazioni (A, B, C, D, E, F come illustrato in fig. 11). Nella fig. 12 è mostrato uno schema di stampa elettrofotografica con scrittura laser.

La superficie del fotoconduttore, realizzata su un tamburo di alluminio, pulita e cancellata nella stazione A da eventuali residui di toner e cariche, viene caricata elettrostaticamente con segno, per esempio, positivo, in modo uniforme nella stazione B così da prepararla a ricevere la scrittura.

La modulazione del laser e la rotazione dello specchio sono controllate da appositi dispositivi in sincronismo con la rotazione del tamburo (posizione angolare), in modo da creare un'immagine latente elettrostatica del carattere sulla superficie fotoconduttiva. Una linea di caratteri viene scritta sfruttando la scansione orizzontale generata dallo specchio ed in verticale usufruendo della rotazione del tamburo. Il numero (n) delle righe di scrittura orizzontali è uguale alla densità dei punti per l'altezza del carattere.

Se la densità dei punti (matrice) è di 300 DPI, sia in orizzontale che in verticale, e l'altezza del carattere è di 2,45 mm il numero delle righe necessarie per scrivere l'intero carattere risulta:

$$n = (300/25{,}4)\ 2{,}45 = 29$$

Non vi è disallineamento orizzontale apprezzabile della linea di stampa data l'alta velocità di scansione orizzontale.

Gli impulsi luminosi, generati dal raggio laser (stazione C), colpiscono la superficie del fotoconduttore creando l'immagine latente elettrostatica del carattere. Alle aree colpite (che assumono un potenziale negativo) si fanno poi aderire delle particelle di toner (inchiostro), caricate elettrostaticamente con segno positivo. A questa operazione di sviluppo provvede la stazione D. Il toner è attratto solo nelle aree negative (immagine latente) ed è respinto dalle altre aree positive (non colpite dal raggio laser) giacenti sulla superficie conduttiva.

L'immagine latente, resa visibile dal toner, è trasferita, da apposita stazione E, su un foglio di carta. La stazione F fissa definitivamente il toner sulla carta.

Le velocità di stampa elettrofotografica vanno da 4 a 150 PPM formato foglio A4, naturalmente con costi crescenti.

La matrice di punti oggi più diffusa è la 300 x 300 DPI.

5. Il colore

La stampa a colori viene eseguita, a secondo delle tecnologie impiegate, con diversi tipi d'inchiostro. Essa è realizzata, di norma, con più passate di stampa.

Gli inchiostri colorati utilizzati sono:

- liquidi per scritture ink jet DOD;
- solidi depositati su nastri di plastica per scritture thermal ink transfer;
- liquidi che impregnano nastri di tessuto per scritture ad aghi;
- toner secchi per scritture elettrofotografiche.

Questi inchiostri, nei tre colori complementari: Cyan (bluverde), Magenta (cremisi) e Yellow (giallo) sono portati, con opportune tecniche, ad interagire con i mezzi di scrittura (aghi, testina termica, ugelli) in modo da essere trasferiti su un foglio di carta. I toner colorati sono, invece, depositati sull'immagine latente elettrostatica realizzata sul fotoconduttore e successivamente trasferiti sul foglio di carta e fissati.

Ricordiamo che la luce bianca si può scomporre in sette colori diversi. Tali colori, detti dell'arcobaleno, possono essere ottenuti con opportune mescolanze dei tre colori primari blu, verde e rosso. Mescolando i colori primari si ottengono i tre colori complementari bluverde, cremisi e giallo.

I vari colori per la stampa sono ottenuti con opportune mescolanze, o per sovrapposizione (più passate di stampa) dei colori complementari. Il nero può essere anche ottenuto sovrapponendo i tre colori complementari Cyan, Magenta e Yellow (sintesi sottrattiva).

È ovvio che per ottenere una buona resa cromatica è fondamentale sovrapporre i vari inchiostri e toner colorati con precisione sul foglio di carta e sul fotoconduttore, per cui i mezzi di scrittura e trasporto foglio devono essere posizionati esattamente.

6. Lo stato dell'arte

In questi ultimi anni le stampanti presenti sul mercato hanno migliorato decisamente le loro prestazioni con un aumento della velocità e qualità di stampa ed una netta diminuzione del rumore acustico (per le tecniche ad impatto), delle dimensioni fisiche, del peso e del costo industriale.

Oggi la progettazione delle stampanti, nei maggiori centri di ricerca mondiali, è realizzata mediante CAD (Computer Aided Design) al fine di simulare nuovi modelli e di ottimizzare i prodotti con l'inserimento di nuovi materiali, componenti e tecniche di produzione. In effetti l'ergonomia, l'operabilità e il design delle stampanti per elaboratori hanno raggiunto oggi livelli più che soddisfacenti.

La fabbricazione si avvale ormai generalmente di robot per assiemare parti/gruppi e di collaudi automatici, per ottenere una diminuzione dei costi ed un più alto e costante livello di qualità del prodotto.

6.1 Tecniche ad impatto

La tecnologia ad aghi è quella oggi maggiormente diffusa nel mercato. Essa soddisfa varie esigenze quali la scrittura di testi con diversi livelli di qualità (differenti densità di punti), la possibilità di fare diagrammi e grafici in genere e di ottenere, contemporaneamente all'originale, un certo numero di copie.
Le altre due tecnologie di stampa presentate (e cioè la tecnica a sfera e a margherita), per il fatto di non fornire scritture per punti non possono essere utilizzate per la grafica; questa limitazione, aggiunta alla bassa velocità ed al maggiore rumore acustico, fa sì che oggi esse siano sempre meno impiegate.

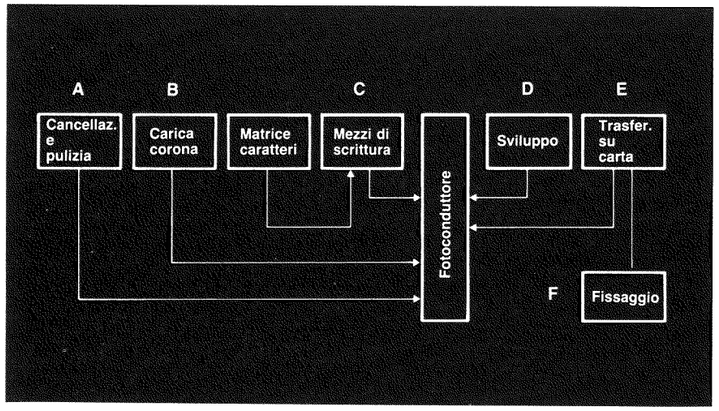


Fig. 11 - Schema a blocchi del processo elettrofotografico.

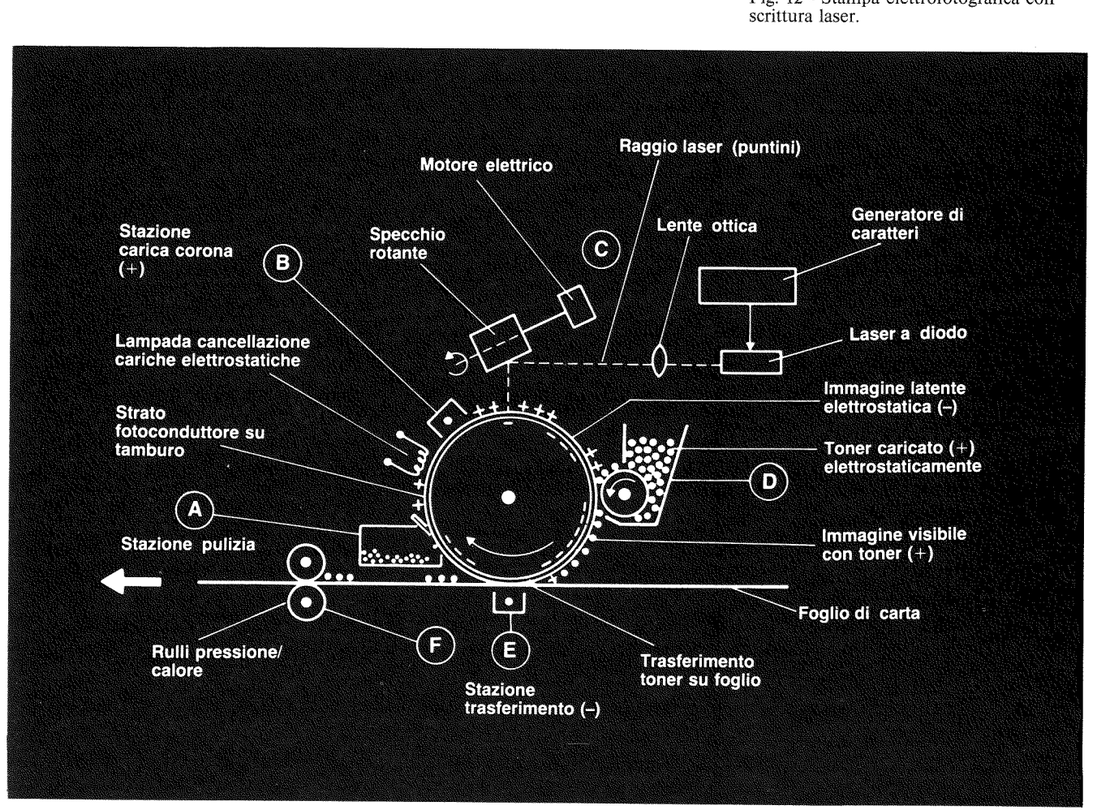


Fig. 12 - Stampa elettrofotografica con scrittura laser.

Recentemente si sono realizzati sensibili miglioramenti nella tecnologia delle testine ad aghi.

Si stanno infatti diffondendo sul mercato testine di scrittura ad aghi a magnete permanente, che differiscono da quelle tradizionali per il fatto che l'ago viene lanciato contro il nastro e la carta mediante una molla a balestra di adeguate caratteristiche. La molla è trattenuta in posizione di riposo precaricata mediante il flusso generato da un magnete permanente.

La scrittura del punto avviene attivando, con opportuno segnale elettrico, un avvolgimento che annulla il flusso magnetico esercitato sulla molla a balestra. In questa situazione la molla, non più trattenuta, spinge, con una determinata energia, l'ago contro il nastro e la carta. A battuta avvenuta, disattivando l'avvolgimento, si ripristina il flusso magnetico che riporta la molla balestra a riposo.

Questa tecnica a magnete permanente permette una maggiore velocità di scrittura: circa 500 CPS. Sono disponibili testine anche a 24 aghi (due file di 12 aghi) per una alta qualità di stampa.

6.2 Tecniche a non impatto

La tecnologia che si espande più rapidamente è quella elettrofotografica con scrittura laser. Infatti con questa tecnica di stampa si possono ottenere scritture per punti e quindi testi, grafici, disegni anche a colori, talora di eccellente qualità. È possibile quindi fare editoria personale o da tavolo con costi e tempi molto contenuti. Inoltre, e questo fa premio, il rumore acustico prodotto da queste stampanti è appena percettibile.
Queste macchine sono molto flessibili, poco ingombranti e con un prezzo appetibile. Hanno varie possibilità di lavoro su fogli di carta con formati UNI diversi, matrici di scrittura superiori a 300 DPI, più cassetti contenenti fogli in entrata ed uscita ed un alto livello di operabilità e affidabilità. In altre parole esse si differenziano alquanto dalle prime macchine apparse sul mercato, ossia sono delle vere stampanti e non delle copiatrici modificate.
L'evoluzione della tecnologia elettrofotografica con scrittura laser è stata e continua ad essere rapida grazie allo sviluppo di nuovi materiali e componenti elettronici ed ottici molto potenti e miniaturizzati.

Le tecnologie Ink Jet DOD e Thermal Ink Transfer hanno ottenuto, in questi ultimi anni, buoni miglioramenti di prestazioni e di affidabilità. Tuttavia le stampanti utilizzanti queste tecnologie sono impiegate in particolari aree applicative con volumi di carta stampata limitati.

6.3 La qualità di stampa

La considerazione principale che si può fare a questo riguardo è che la scrittura per punti, ove sia realizzata con una densità di punti sufficientemente elavata, può fornire risultati che non hanno niente da invidiare a quelli ottenuti con le stampanti Solid Font.
A conferma di ciò, la fig. 13 mostra un campione di scrittura ottenuto da una stampante a impatto (testina a 9 aghi, densità 100 x 144 DPI). La fig. 14 fornisce un altro esempio di scrittura per punti generata, invece, con tecnica a non impatto (stampante laser, densità 300 x 300 DPI).
Le stampanti a punti presentano, inoltre, la prerogativa delle capacità grafiche, che consentono la stampa di immagini, quale può essere per esempio una fotografia, con ottimi risultati.
Un'idea di tali capacità è data dalla fig. 15, ottenuta con la stessa stampante di cui alla fig. 14.

Fig. 13 - Esempio di scrittura per punti con tecnica a impatto (stampante 4/41 della Honeywell Bull, con testina a 9 aghi e densità 100 x 144 DPI).

```
        La ridotta profondità e l'innovativo bottom feed del-
la 4/41, ne consentono  l'impiego  in spazi limitati, senza ri-
chiedere supporti specifici.
        Un display alfanumerico  posto sul pannello operatore
consente l'immediata e facile selezione delle caratteristiche e
parametri di stampa.
        La 4/41 offre inoltre una ricca gamma di prestazioni,
quale per esempio  una ampia scelta di font e codici a barre.
        Il colore è standard.
```

Pochi sviluppi nel campo dell'informatica hanno attirato l'attenzione degli utenti come il desk top publishing.

(Courier 10 cpi)

In sostanza, il desk top publishing da' all'utente il totale controllo del processo di creazione del documento stampato.

(Courier 10 cpi italic)

Fornisce un metodo per la produzione di documenti dal taglio e dall'aspetto professionale senza bisogno di ricorrere all'esterno per costosi procedimenti di fotocomposizione e grafici.

(Prestige Elite 12 cpi)

Tramite questo sistema si possono scrivere dei testi, disegnare illustrazioni, incorporare fotografie, progettare layout di stampa e stampare un documento finito per mezzo di una stampante laser.

(Compresso 16.7 cpi)

I sistemi " desk top publishing" consentono un decremento del costo di produzione dei documenti e un incremento della flessibilita' di produzione sia in termini di tempo sia in termini di risorse.
(Times Roman proportional)

Fig. 14 - Esempio di scrittura con tecnica non a impatto (stampante "*Laser Page 801*" della Honeywell Bull, con scrittura laser 300 x 300 DPI).
Le scritte tra parentesi fanno riferimento allo stile dei caratteri e alla loro densità (cpi = caratteri per inch).

6.4 Indici di costo

Per dare una indicazione di massima di quanto variano fra loro i costi di stampa delle tecnologie presentate, è stata compilata la tabella di fig. 16.

Si precisa che, per semplificare il conteggio, è stato calcolato solo il consumo dei ricambi, quali nastri, inchiostri, toner, fotoconduttori, ecc necessari per stampare in nero 1.273 caratteri font Pica 10 (pagina standard Honeywell Bull) su un foglio di carta UNI-A4. Non è stato possibile, perchè le tecnologie sono diverse fra loro, omogeneizzare i costi di stampa con un unica matrice di punti o qualità di stampa. Pertanto gli indici di costo forniti sono medi ed in armonia con la qualità di stampa generata da ogni singola tecnica.

Il costo del foglio è escluso poichè esso è legato alla qualità della carta che si vuole utilizzare.

È stata scelta come tecnica di riferimento la scrittura ad aghi, per la sua grande diffusione, per cui il suo indice è a quota 100.

Fig. 15 - Esempio di stampa di immagini con tecnica elettrofotografica (stessa stampante della fig. 14).

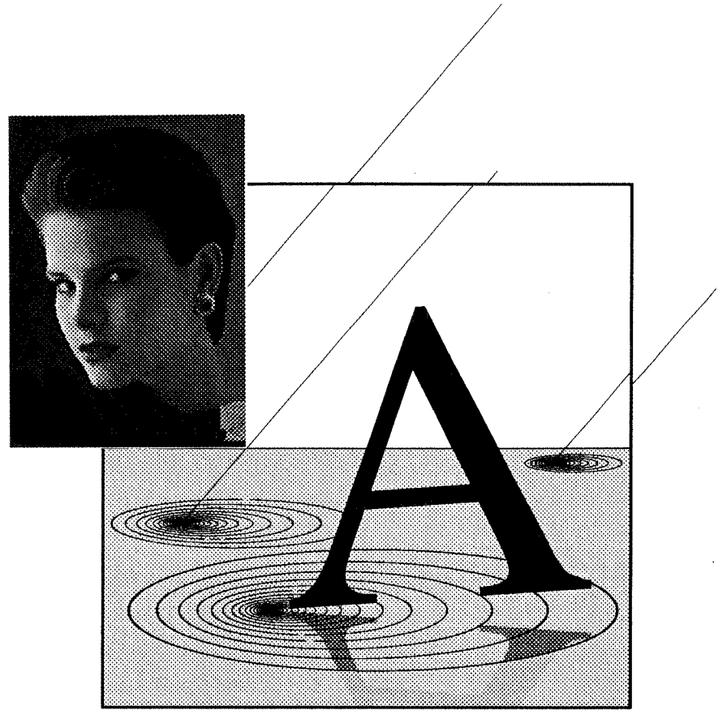


Fig. 16 - Indici medi del costo di stampa delle tecnologie presentate nell'articolo. (Si tenga presente che le velocità di stampa sono tra loro diverse).

TECNICHE DI STAMPA	1.273 CARATTERI STAMPATI IN NERO SU FOGLIO A4 CON:	INDICE MEDIO DEL COSTO DI STAMPA
9 AGHI	60 x 72 DPI	100
24 AGHI	180 x 180 DPI	220
SFERA	CARATTERE PIENO	250
MARGHERITA	CARATTERE PIENO	270
INK JET DOD	180 x 180 DPI	620
THERMAL INK TRASFER	360 x 240 DPI	2400
ELETTROFOTOGRAFICA CON SCRITTURA LASER	300 x 300 DPI	750

7. Conclusioni

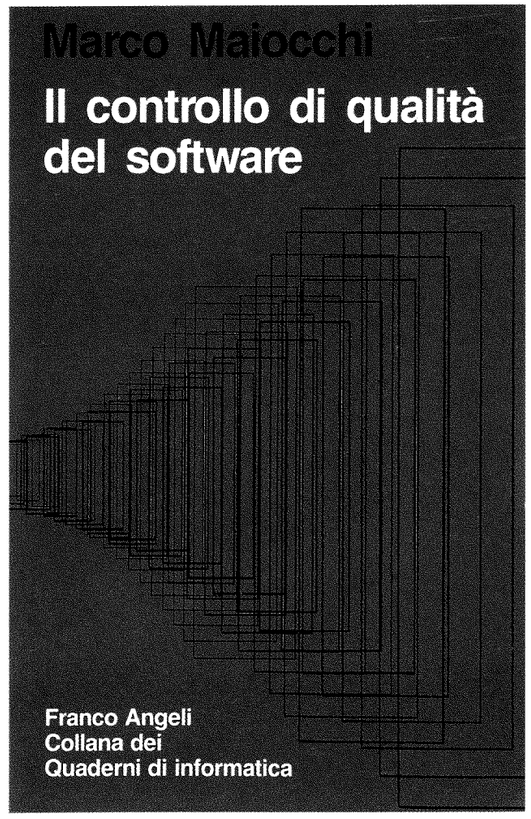
Con la grande diffusione in atto dei Personal Computer nelle applicazioni Home, Office, Electronic Publishing, le stampanti seriali avranno ancora per molti anni un grande mercato. La tecnologia elettrofotografica acquisterà più posizioni, invadendo fasce applicative che alcuni anni fa erano riservate alla tecnica ad aghi. Tuttavia, anche per le stampanti ad aghi si prevede un rafforzamento dovuto ai progressi in termini di velocità, qualità di stampa e rumore acustico.

Traguardando un futuro più lontano, indubbiamente prenderà sempre più piede la utilizzazione di supporti elettronici per memorizzare le informazioni. Ma la carta stampata manterrà comunque sempre un suo ruolo.

BIBLIOTECA NOVITÀ

I TESTI DEI **QUADERNI DI INFORMATICA**

Marco Maiocchi: Il controllo di qualità del software
Collana dei Quaderni di Informatica, Ed. F. Angeli, 1988 (pag. 290, L. 30.000)



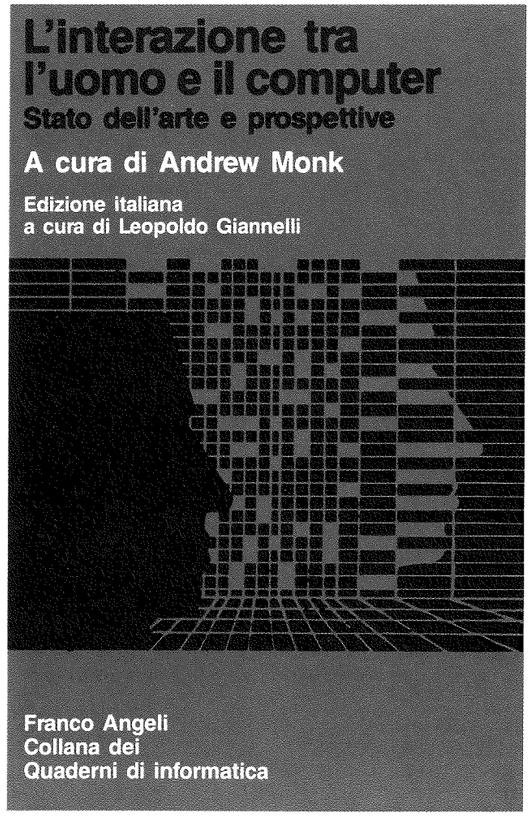
La crescita di dimensione, di complessità e di diffusione del software ha già da tempo mostrato la necessità di un suo accurato Controllo della Qualità.
Tale attività, presente in ogni settore industriale di produzione, tarda a consolidarsi nella produzione del software, in gran parte a causa di limitate competenze ed elevati costi, cosicché spesso i programmi in uso presentano malfunzionamenti anche gravi, che rischiano di arrecare danni molto superiori ai costi di un'attività di qualificazione. Inoltre, la qualificazione di un prodotto software tende a individuare le cause di una cattiva qualità presente nei prodotti, influendo così sui metodi di produzione, e quindi migliorando il rapporto costo/beneficio dello sviluppo stesso.
Questo libro si rivolge a tutti gli operatori coinvolti nella produzione di software, sia come responsabili diretti dello sviluppo, sia come funzione di controllo, fornendo un quadro completo di come collocare il Controllo di Qualità nella realizzazione di programmi, di quali strategie e metodologie di qualificazione adottare, di quali tecniche applicare e di come orientare la produzione del software a «produrre programmi di qualità» fin dall'inizio.
Il testo, pur dedicando significativo spazio agli aspetti metodologici e scientifici teorici, esamina con cura gli aspetti più propriamente aziendali di costi e benefici, procede con ricchezza di esempi e illustrazioni di strumenti per la qualificazione, che lo rendono uno strumento di formazione e una valida guida operativa.

Marco Maiocchi, laureato in Fisica, è docente di Teoria e Applicazioni delle Macchine Calcolatrici all'Università di Milano, presso cui svolge da oltre quindici anni ricerche applicate sui temi dell'ingegneria del Software, sul Controllo di Qualità e sull'Assicurazione di Qualità del Software. È uno dei fondatori della Etnoteam, leader nel settore, con cui collabora attivamente. È autore di numerosissimi libri e articoli scientifici di livello nazionale e internazionale relativi ai temi di cui si occupa.

BIBLIOTECA NOVITÀ

I TESTI DEI **QUADERNI DI INFORMATICA**

A. Monk et al.: L'interazione tra l'uomo e il computer.
Ediz. ital. a cura di Leopoldo Giannelli
Collana dei Quaderni di Informatica, Ed. F. Angeli, 1988 (pag. 315, L. 34.000).



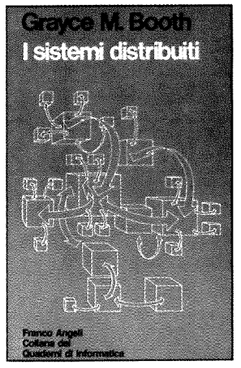
Man mano che cresce lo spazio occupato dal computer nella nostra vita, acquista maggior importanza il grado d'accettabilità, d'efficienza e d'attrazione dei sistemi su di esso basati. Per l'utente finale - l'uomo della strada che con essi deve conviveve - il punto è di migliorare la qualità della propria vita e, contemporaneamente, sfruttare le possibilità che i nuovi sistemi gli aprono nel campo professionale. Per le società di hardware e di software la conquista di nuove quote di mercato è sempre più subordinata alla capacità di costruire e vendere prodotti interattivi, amichevoli, a misura umana. E questo apre nuove, affascinanti sfide alla stessa conoscenza dell'uomo: della sua percezione visiva, del suo modo di pensare e ragionare, della sua memoria, della comunicazione uomo-macchina, dei modelli mentali adottati dagli «esperti» dei vari campi per affrontare e risolvere i diversi problemi (curare un malato, progettare un edificio, costruire un prodotto, ecc.).
Questo volume, scritto da studiosi di discipline diverse, offre una visione sintetica delle conoscenze fino ad oggi acquisite sull'interazione uomo-computer, sia con ricerche di laboratorio che attraverso sperimentazioni sul campo.
L'obiettivo principale è di dare organicità, in un unico quadro generale, a un sapere che deve entrare a far parte del patrimonio professionale di tutti coloro che operano, a qualsiasi titolo e livello, nel campo dell'informatica, della organizzazione e della gestione del personale; vale a dire il progettista, l'analista e il programmatore di sistemi informatici, ma anche lo psicologo, l'esperto di ergonomia e l'industrial designer.

I TESTI DEI
QUADERNI DI INFORMATICA

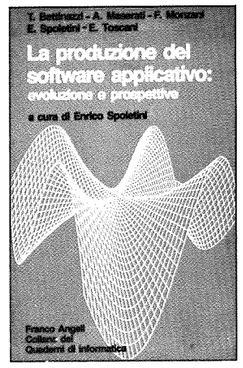


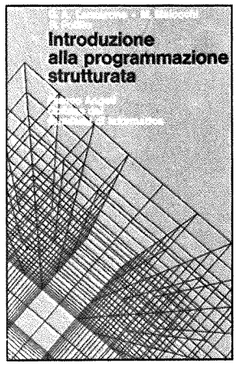


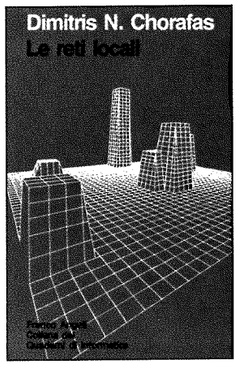












I TESTI DEI
QUADERNI DI INFORMATICA

